

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# **Návrh a realizace řídicí aplikace pro PLC s využitím konceptu digitálního dvojčete**

## **Design and Implementation of PLC Based Control Application using Digital Twin Concept**

## Zadání bakalářské práce

Student: **Filip Vicherek**  
Studijní program: B2649 Elektrotechnika  
Studijní obor: 2612R041 Řídicí a informační systémy  
Téma: Návrh a realizace řídicí aplikace pro PLC s využitím konceptu  
digitálního dvojčete  
Design and Implementation of PLC Based Control Application  
using Digital Twin Concept  
Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Analýza možností návrhu a testování řídicích aplikací s využitím konceptu digitálního dvojčete.
2. Realizace digitálního dvojčete vybraného laboratorního systému s využitím nástroje Tecnomatix Process Simulate.
3. Návrh a realizace řídicí aplikace pro vybraný laboratorní systém.
4. Testování řídicí aplikace pomocí digitálního dvojčete.
5. Ověření jednotlivých variant připojení řídicího systému k digitálnímu dvojčeti.
6. Zhodnocení výsledků řešení.

### Seznam doporučené odborné literatury:


- [1] GREENGARD, Samuel. *The internet of Things*. Massachusetts: MIT Press, 2015. ISBN 978-0-262-52773-6.  
[2] BERGER, Hans. *Automating with SIMATIC*. 5th edition. Erlangen, Germany: Publicis Publishing, 2013, 284 p. ISBN 978-3895783876.  
[3] Technická dokumentace k systému Simatic.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **doc. Ing. Jiří Koziorek, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

  
\_\_\_\_\_  
doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry



  
\_\_\_\_\_  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 26. dubna 2019

  
.....

Rád bych na tomto místě poděkoval vedoucímu práce doc. Ing. Jiří Koziorkovi, Ph.D za odborné vedení, pomoc a rady při zpracování mé bakalářské práce. Dále celé své rodině, která mě podporovala a poskytla mi prostor pro realizaci této práce.

## **Abstrakt**

Bakalářská práce se zabývá návrhem a tvorbou řídicí aplikace pro digitální dvojče výrobní linky vytvořené v programu Tecnomatix Process Simulate. V této práci je provedena analýza a testování návrhu řídicí aplikace za pomoci tohoto digitálního návrhu. Vytvořený digitální návrh linky obsahuje tři statická robotická ramena a jedno robotické rameno na pojezdu. Navržená linka skládá výrobky z LEGO kostek. Řídicí aplikace je navržena pro PLC v jazyku ladder. V práci je popsáno jednotlivé připojení PLC pomocí jednotlivých OPC serverů a simulátoru PLC k digitálnímu návrhu linky.

**Klíčová slova:** bakalářská práce, digitální dvojče, Tecnomatix Process Simulate, TIA portál, ladder, OPC DA, OPC UA, PLCSIM Advanced v2.0

## **Abstract**

The bachelor thesis deals with design and creation of control application for digital twin production lines created in Tecnomatix Process Simulate. In this work is done analysis and testing of control application design using this digital design. The created digital line design includes three static robotic arms and one robotic arm on the traverse. The proposed line consists of products from LEGO bricks. The control application is designed for PLC ladder language. The thesis describes individual PLC connection using individual OPC servers and PLC simulator for digital line design.

**Key Words:** Bachelor thesis, digital twin, tecnomatix Process Simulate, TIA portal, ladder, OPC DA, OPC UA, PLCSIM Advanced v2.0

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>8</b>
<b>Seznam obrázků</b>	<b>9</b>
<b>Seznam tabulek</b>	<b>11</b>
<b>Úvod</b>	<b>12</b>
<b>1 Návrh a testování řídicího systému s využitím digitálního dvojčete</b>	<b>14</b>
1.1 Návrh řídicí aplikace za pomoci digitálního dvojčete . . . . .	14
1.2 Testování řídicí aplikace za pomoci digitálního dvojčete . . . . .	15
1.3 V-model . . . . .	16
<b>2 Vytvořené digitální dvojče</b>	<b>18</b>
2.1 Práce v Tecnomatix Process Simulate . . . . .	18
2.2 Vytvořené modely . . . . .	19
2.3 Generování a život produktu . . . . .	20
2.4 Dopravníky . . . . .	20
2.5 Robotická ramena a operace . . . . .	23
2.6 Senzory . . . . .	31
<b>3 Funkční analýza</b>	<b>33</b>
3.1 Dopravníky . . . . .	33
3.2 Robotická ramena . . . . .	35
3.3 Senzory . . . . .	37
<b>4 Řídicí aplikace</b>	<b>39</b>
4.1 Dopravníky . . . . .	39
4.2 Robotická ramena . . . . .	41
<b>5 Připojení digitálního dvojčete</b>	<b>46</b>
5.1 OPC . . . . .	46
5.2 PLCSIM Advanced . . . . .	49
5.3 Nastavení připojení . . . . .	49
<b>6 Testování</b>	<b>51</b>
<b>7 Závěr</b>	<b>52</b>
<b>Literatura</b>	<b>54</b>

<b>Přílohy</b>	<b>54</b>
<b>A Seznam příloh na přiloženém CD</b>	<b>55</b>

## Seznam použitých zkratek a symbolů

TPS	– Tecnomatix Process Simulate
CAD	– Computer-Aided Design
PLC	– Programmable Logic Controller
OPC	– Open Platform Communications
OPC DA	– OPC Data Access
OPC UA	– OPC Unified Architecture
LAD	– Ladder Diagram
IP	– Internet Protocol
OS	– Operační Systém
OLP	– Off-line programming
SOAP	– Simple Object Access Protocol
HTTP	– Hypertext Transfer Protocol
TCP/IP	– Transmission Control Protocol/Internet Protocol
HMI	– Human Machine Interface
SCADA	– Supervisory Control And Data Acquisition



## Seznam obrázků

1	Model úrovní digitálního dvojčete . . . . .	15
2	V-Model . . . . .	16
3	Vytvořená linka . . . . .	18
4	Popis linky . . . . .	19
5	Části boudy . . . . .	21
6	Složená bouda . . . . .	22
7	Vygenerovaný materiál . . . . .	22
8	Ukázka logického bloku . . . . .	23
9	Vývojový diagram generování materiálů . . . . .	23
10	Tok materiálu . . . . .	24
11	Nastavení dopravníku . . . . .	25
12	Směry a názvy dopravníků . . . . .	25
13	Výpis pozic . . . . .	26
14	Robotická operace . . . . .	26
15	OPL menu . . . . .	27
16	Spojování objektů pomocí operací . . . . .	27
17	Robotické rameno R1 . . . . .	28
18	Robotické rameno R3 . . . . .	29
19	Robotické rameno R4 . . . . .	30
20	Vytváření a parametrování externí osy . . . . .	30
21	Nastavení senzorů . . . . .	32
22	Rozmístění senzorů . . . . .	32
23	Funkční blok pro Conv_gen . . . . .	39
24	Funkční blok pro jednosměrné dopravníky . . . . .	40
25	Funkční blok pro obousměrné dopravníky . . . . .	41
26	Funkční blok pro řízení křižovatek T1-T3 . . . . .	42
27	Funkční blok pro testování . . . . .	43
28	Funkční blok pro řízení křižovatky T4 . . . . .	43
29	Funkční blok pro robotická ramena R1-R3 . . . . .	44
30	Funkční blok pro robotické rameno R4 . . . . .	45
31	Možnosti řízení v TPS . . . . .	46
32	Průběh komunikace mezi TPS a PLC . . . . .	47
33	Externí připojení . . . . .	47
34	Nastavení OPC DA . . . . .	48
35	Nastavení KEPServeru . . . . .	48
36	Připojení k PLCSIM Advanced . . . . .	49
37	Nastavení PLCSIM Advanced . . . . .	50

38	Nastavení signálů . . . . .	50
39	Simulation panel . . . . .	51

## Seznam tabulek

1	Tabulka robotických signálů . . . . .	27
2	Čísla robotických operací pro R1 a R2 . . . . .	28
3	Čísla robotických operací pro R3 . . . . .	28
4	Čísla robotických operací pro R4 . . . . .	31
5	Tabulka výstupních signálů pro řízení dopravníků . . . . .	36
6	Výstupní signály pro řízení robotických ramen . . . . .	37
7	Vstupní signály pro řízení robotických ramen . . . . .	38
8	Vstupní signály ze senzorů . . . . .	38

## Úvod

Tato bakalářská práce se zabývá analýzou a možnostmi návrhu řídicí aplikace s využitím konceptu digitálního dvojčete. Dále se zabývá samotným návrhem digitálního dvojčete v programu Tecnomatix Process Simulate. Vytvořené digitální dvojče se skládá ze čtyř robotických ramen, jedno z ramen je postavené na pojezdu. Zabývá se návrhem a realizací řídicí aplikace pro vytvořené digitální dvojče a testováním jednotlivých možností připojení PLC k dvojčeti. V závěru je provedena analýza možností testování na digitálním dvojčeti.

V první kapitole bakalářské práce je popsána problematika digitálního dvojčete, jeho výhody a využití. Teoretický popis tvorby digitálního dvojčete. Využití digitálního dvojčete spolu s tvořeným řídicím systémem a možnostmi návrhu a testování řídicího systému s dvojčetem. Popis vybraného modelu pro návrh a realizaci řídicí aplikace, zvolen byl V-model.

Druhá kapitola popisuje realizaci vytvořené linky v prostředí Tecnomatix Process Simulate. Popisuje základní práci v TPS, rozdělení objektů a popis produktu, který se skládá z LEGO kostek. Dále zahrnuje nastavení generování materiálu, tvorbu dopravníků, vytvoření signálu pro řízení dopravníků a robotických ramen. Robotická ramena a jejich operace jsou popsány v této kapitole. Popis druhů senzorů a jejich využití v lince. Výpis jednotlivých operací, které budou následně využity pro tvorbu řídicí aplikace.

Třetí kapitola se zabývá návrhem řídicí aplikace pro vytvořenou linku, který je popsán funkční analýzou, která popisuje veškerou problematiku linky. Funkční analýza je rozdělena do částí, ve kterých jsou popsány dopravníky, robotická ramena a senzory. Výstupem funkční analýzy jsou vstupní a výstupní signály, které budou využity pro připojení reálného PLC a tvorbu řídicí aplikace.

Čtvrtá kapitola popisuje vytvořenou řídicí aplikaci pro PLC. Tato aplikace je napsána v programovacím jazyce LAD (ladder), za použití programu TIA portál v14. V kapitole jsou popsány řídicí algoritmy, které slouží k řízení jednotlivých částí linky, jako jsou čtyři dopravníkové křižovatky, jednosměrné a obousměrné dopravníky a robotická ramena. Řídicí algoritmy jsou vytvořené ve funkčních blocích, pro jednoduchou možnost jejich kopírování a přehlednější provedení kódu.

V páté kapitole jsou uvedeny možnosti připojení Tecnomatix Process Simulate k fyzickému PLC od firmy SIEMENS řady S7-1500. Popsány jsou obě OPC komunikace OPC DA a OPC UA. V kapitole je napsán postup, jak se vytvoří připojení k těmto komunikacím. Pro vytvoření serveru OPC DA komunikace byl zvolen software KEPServerEX v6, v kapitole je uveden postup vytvoření tohoto serveru. Poslední otestovanou možností připojení bylo využití simulovaného PLC. Instance tohoto PLC byla vytvořena použitím softwaru PLCSIM ADVANCED V2.0, ke které se následně Tecnomatix Process Simulate připojil. V této části je vytvořen postup vytvoření instance PLC a následné připojení TPS k této instanci.

Poslední kapitola se zabývá testováním digitálního dvojčete a ověření správné funkčnosti jednotlivých komunikací. V této části bylo odstraněno mnoho chyb z řídicí aplikace. Výsledkem

této kapitoly jsou videa zprovozněné linky, která byla řízena pomocí reálného PLC. Video se nacházejí v příloze.

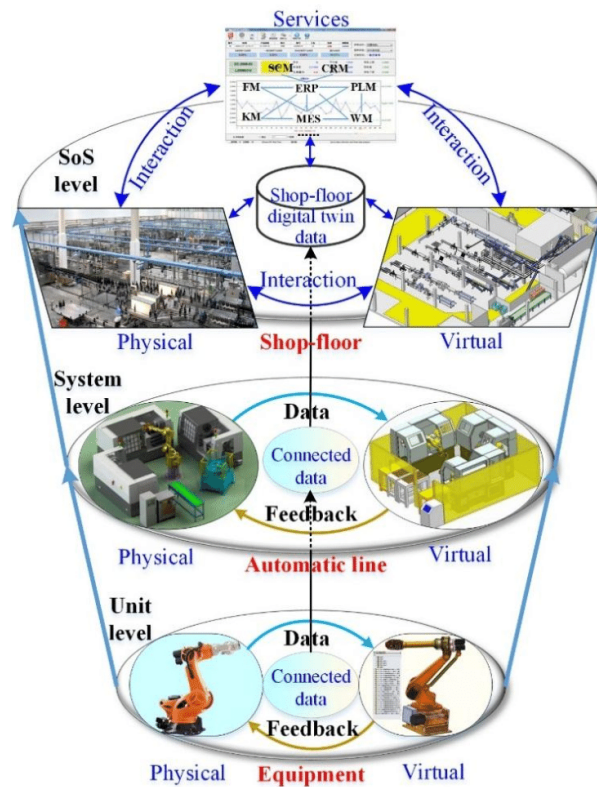
# 1 Návrh a testování řídicího systému s využitím digitálního dvojčete

Digitální dvojče je klon, který se vztahuje k reálným věcem, které jsou vytvořeny ve virtuálním prostředí. Digitální model může být vytvořen k již stávajícímu zařízení, nebo se vytváří spolu s vytvářeným strojním zařízením a zůstávají propojeni po celou dobu svého životního cyklu. Díky tvořenému digitálnímu dvojčeti lze odhalit chyby jak v konstrukčním návrhu, tak řídicím návrhu. Doposud se používal postup, při kterém se vytvořil nákladný prototyp, který vznikl z plánů, výkresů, zkušeností konstruktérů a programátorů. Dále pomáhá s optimalizací výrobního procesu tvořeného strojního zařízení. Další velkou výhodou použití digitálního dvojčete je snížení výrobního času pro dokončení a otestování strojního zařízení a ušetření výrobních nákladů. Poskytuje možnost sbírání dat o probíhajícím výrobním procesu a toku materiálů, např. počty vyrobených kusů, poruchovost. Tyto data lze vyhodnocovat pomocí speciálních nástrojů a výsledky využít pro optimalizaci výrobních postupů. Oba modely se navzájem ovlivňují z nasbíraných dat. [1] [2]

Digitální dvojče lze rozdělit na tři úrovně, a to jednotkovou úroveň (Unit level), úroveň systému (System level) a systém systémů (System of system). První nejnižší úroveň je jednotková. Digitalizují se zde nejmenší části výrobního procesu (např. robotické ramena, dopravníky). Další úrovní je systémová úroveň. Do této úrovně spadají celé výrobní linky. Více těchto digitálních dvojčat vytvářejí poslední úroveň a to systém systému. Poslední úroveň si lze představit jako továrnu plnou digitálních dvojčat, které navzájem spolupracují. Digitální dvojčata jsou připojena k fyzickým zařízením. Fyzická zařízení posílají data do digitálního dvojčete, které řídí dvojče, zároveň toto dvojče dokáže ovlivňovat fyzický model z přijatých dat. Veškerá nasbíraná data jsou zpracována pomocí nadřazených řídicích systémů. Nadřazené řídicí systémy poté ovlivňují digitální dvojčata a fyzické systémy. Na obrázku 1 lze vidět jednotlivé fyzické a digitální úrovně dvojčete. [4]

## 1.1 Návrh řídicí aplikace za pomoci digitálního dvojčete

Návrh řídicí aplikace je složitý proces, existuje mnoho modelů pro jeho návrh. Jednotlivé modely se skládají z více částí, které řeší jednotlivé problémy při návrhu. Výsledný návrh se odvíjí od kvality zpracování jednotlivých částí. Neexistuje nijak normovaný postup návrhu, většina firem má ten svůj. Většina programátorů je pod časovým nátlakem při opravování chyb s blížícím se termínem předání zakázky. S tímto problémem by mohlo pomoci digitální dvojče, díky kterému lze začít testovat řídicí aplikaci dříve. Při tvorbě digitálního dvojčete a řídicího systému je nutné dokonale znát požadavky zákazníka. Tyto požadavky vedou k vytvoření funkční analýzy, elektrického a mechanického provedení zařízení. S touto analýzou jsme schopni začít budovat digitální dvojče s co největší přesností k reálnému návrhu. Digitální dvojče lze rozdělit na menší části. První částí budování dvojčete je sestavení všech mechanických komponent v CAD ná-



Obrázek 1: Model úrovní digitálního dvojčete [4]

strojích v reálných souřadnicích. Poté je nutné vytvořené komponenty exportovat do programu pro tvorbu digitálního dvojčete a vytvořit seznamy produktů, materiálu, procesů, zdrojů. Dalším krokem je přidání kinematiky pro vložené mechanické prvky, vytvoření trajektorií pro jednotlivé mechatronické části. Přidáním senzorů a akčních členů do mechanických prvků zajistí možnost řízení a zpětné vazby. Digitální dvojče si lze rozdělit na části, pro které se může vytvářet vytvářet řídicí aplikaci. Při tvorbě řídicí aplikace jsou využity technologické signály z digitálního dvojčete. Rozdělením řídicí aplikace na jednotlivé segmenty např. (řízení dopravních systému, řízení robotů, atd.) umožní jednodušší a přehlednější tvorbu aplikace. [2] [3]

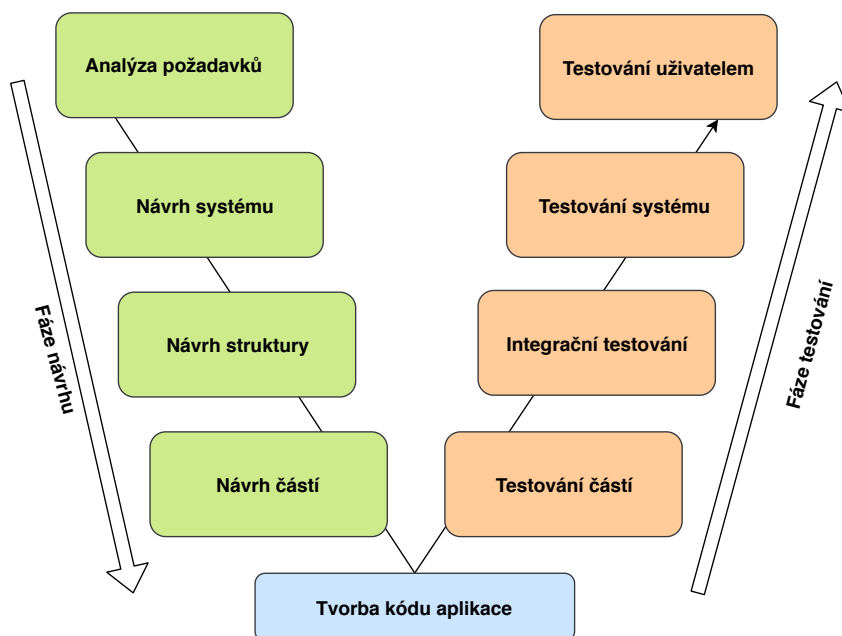
## 1.2 Testování řídicí aplikace za pomoci digitálního dvojčete

Testování pomocí digitálního dvojčete má mnoho výhod, ale i nevýhod. Velkou výhodou tvorby řídicí aplikace spolu s tvorbou digitálního dvojčete je možnost začít s testováním dříve. Důležité je ověření správné komunikace mezi digitálním dvojčetem a řídicím systémem, lze využít softwary, které nám zobrazí stavy sledovaných signálů. Během testování řídicí aplikace, lze přijít na mnoho chyb v stávajícím digitálním návrhu. Tyto chyby lze jednoduše odstranit bez velkých finančních a časových ztrát oproti stávajícímu testování. Ošetření kolizních stavů robotů. Možnost testovat navržené HMI pro navrhované zařízení. Testování na digitálním dvojčeti nemusí

úplně odpovídat testování na fyzickém modelu. Fyzický model se může chovat trochu odlišně od toho digitálního, proto je potřeba provádět testování na reálném modelu a provádět úpravy na digitálním dvojčeti z výsledků testování. [2]

### 1.3 V-model

Jeden z mnoha postupů při tvoření řídicí aplikace. Model se sekvenčním přístupem, který je rozdělen na dvě strany obrázek 2. Levá strana rozkládá systém na části, ve kterých je provedena analýza celého systému a návrhy jednotlivých částí. Pravá strana se zabývá testováním jednotlivých částí z levé strany. V-model je ideální pro projekty, kde se dodržují přísné časové podmínky pro dodání produktu. S postupem, který V-model vytváří je jednodušší vytvořit časový rozvrh pro jednotlivé části návrhu, testování a oprav. Nevýhoda V-modelu je v neschopnosti se přizpůsobit na velké změny v době návrhu. [5]



Obrázek 2: V-Model

#### Fáze návrhu:

- **Analýza požadavků** - Jedná se o první fázi vývojového cyklu, ve které jsou požadavky produktu chápány z pohledu zákazníka. Tato fáze zahrnuje detailní komunikaci se zákazníkem s cílem pochopit jeho očekávání a přesné požadavky.
- **Návrh systému** - Jakmile jsou vypracovány požadavky na produkt, je potřeba navrhnout celý systém. Tento návrh bude mít podrobný popis kompletního hardwarového a komunikačního nastavení produktu, který je ve vývoji. Plán testů systému je vyvinut na základě návrhu systému. Pokud to provedete v dřívější fázi, ponechá více času na provedení vlastního testu později.



- **Návrh struktury** - Přenos dat a komunikace mezi interními moduly a okolním světem (jinými systémy) je v této fázi jasně pochopena a definována. S těmito informacemi lze během této fáze navrhnout a zdokumentovat integrační testy. Označován jako design na vysoké úrovni.
- **Návrh částí** - V této fázi je specifikován podrobný interní návrh pro všechny systémové části, označován jako design na nízké úrovni. Je důležité, aby byl design kompatibilní s ostatními částmi architektury systému a dalšími externími systémy.

#### **Programování:**

- **Tvorba kódu aplikace** - Kódování systémových modulů napomáhají vytvořené předchozí návrhy. Nejvhodnější programovací jazyk je založen na systémových a architektonických požadavcích. Kódování se provádí na základě pokynů a standardů kódování.

#### **Fáze testování:**

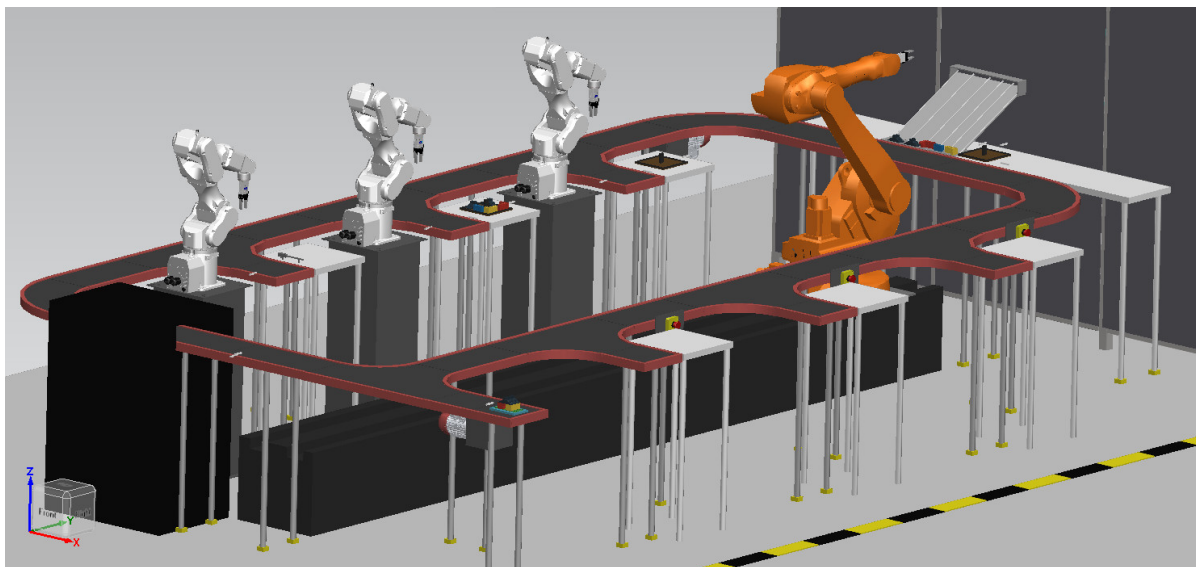
- **Testování částí** - Testování navržených částí jsou prováděny na kódu během této ověřovací fáze. Jednotkové testování je testování na úrovni kódu a pomáhá odstranit chyby v raném stádiu, i když všechny chyby nemohou být odhaleny jednotkovým testováním.
- **Integrační testování** - Integrační testování je spojeno s fází architektonického návrhu. Integrační testy jsou prováděny za účelem testování komunikace interních částí v systému.
- **Testování systému** - Testování systému je přímo spojeno s fází návrhu systému. Systémové testy kontrolují funkčnost celého systému a komunikaci vyvíjeného systému. Většina problémů s kompatibilitou softwaru a hardwaru může být odhalena během provádění tohoto testu systému.
- **Testování uživatelem** - Uživatelské testování je spojeno s fází analýzy obchodních požadavků a zahrnuje testování produktu v uživatelském prostředí. Testy odhalují problémy s kompatibilitou s ostatními systémy dostupnými v uživatelském prostředí.

## 2 Vytvořené digitální dvojče

Digitální dvojče bylo navrženo pro projekt Smart Factory ve škole. Vytvořená linka obsahuje čtyři robotická ramena, z toho první dvě (R1 a R2) budou sestavovat výrobky nezávisle na sobě, třetí (R3) bude provádět testování a složitější úpravy na výrobcích. Čtvrté robotické rameno (R4) je umístěné na pojezdu, které se bude starat o nakládání materiálu ze skladiště na táb a jeho doručení k jednotlivým pracovištím. Na obrázku 3 lze vidět vytvořené dvojče a na obrázku 4 je linka rozkreslena blokově. Pohyb tvořeného produktu je realizován pomocí paletky, která se pohybuje po dopravnících.

Generování materiálů probíhá na blistrech, kde jednotlivé části montovaného výrobku sjedou na dojezdovou plochu, kde setrvávají do vyzvednutí robotickým ramenem R4. Generován je i táb, pomocí kterého je přepravován materiál.

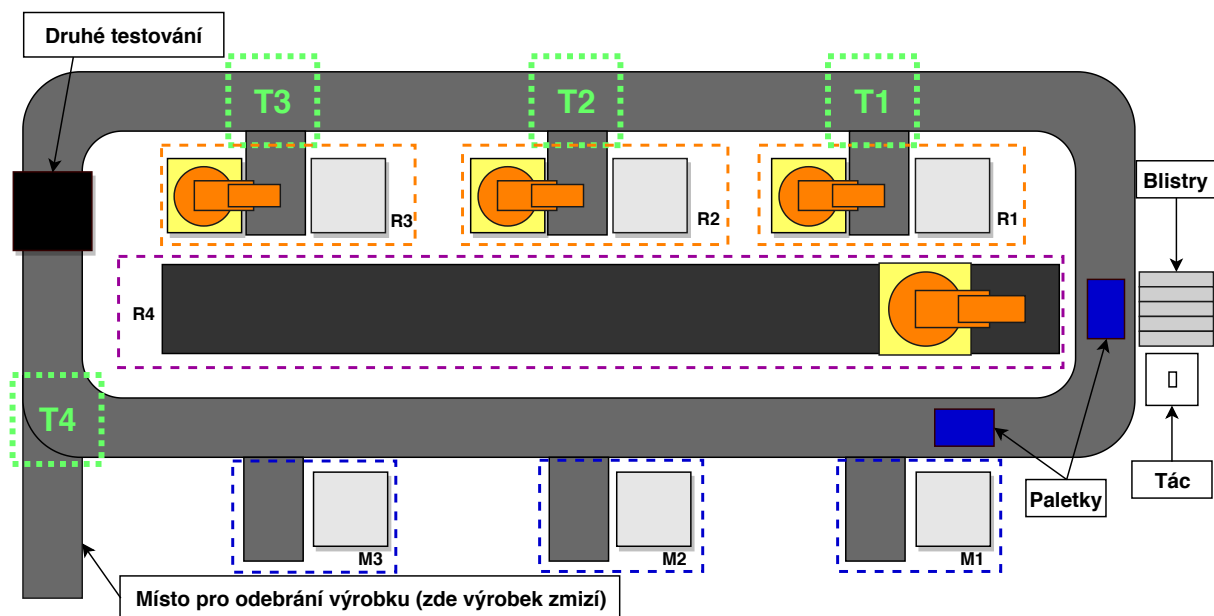
Testovací stanice slouží k otestování zpravené funkčnosti vyrobeného výrobku. Pro účely simulace se v něm paletka s hotovým výrobkem na chvíli zastaví a poté bude pokračovat k odezdání. Dále se zde nacházejí tři manuální pracoviště, které budou sloužit pro manuální stavění výrobků. V dalších kapitolách je popsána tvorba této linky v programu Tecnomatix Process Simulate, dále jen TPS. Veškeré informace ohledně funkčnosti prvků v TPS (dopravníků, robotických ramen), byly čerpány z *Helpu* a fóra. [6] [7]



Obrázek 3: Vytvořená linka

### 2.1 Práce v Tecnomatix Process Simulate

Nejdůležitější částí je při vytváření projektu nastavení jeho cesty na disku. Tato cesta se musí dodržovat i po přenesení návrhu na jiný počítač. Jednotlivé vytvořené a vkládané objekty v TPS musí mít také zachovanou stejnou adresu uložení nebo vložení. Jestliže se změní cesta uložení



Obrázek 4: Popis linky

některé z komponent, TPS ji není schopen nalézt a komponenta zmizí z vývojového prostředí. Cesta vytvořené linky C:\PLM\Process\_Simulate\_data\Projects\VSB\_Virtual\_Twin. Netvořit názvy s českými znaky, při přechodu na operační systém s jiným jazykem může vytvořit problémy.

Důležitou funkcí TPS jsou dva pracovní režimy *Standart Mode* a *Line Simulation Mode*.

- **Standart Mode** - slouží k vkládání, modelování a posunu objektů, vytváření a úpravu operací.
- **Line Simulation Mode** - spouštění a provádění úprav jednotlivých simulací, spouštění celé simulace, práci se signály, vytváření toků materiálů.

## 2.2 Vytvořené modely

V TPS lze modelovat různé objekty a nástroje, ale modelování je velice jednoduché a nenabízí takové možnosti, jako jiné programy, které se přímo věnují této problematice. Na objekty jako je stůl a podstava pro dopravník bohatě postačilo modelování v TPS. Pro tvorbu dalších objektů byl použit program Inventor 2019, ve kterém je vymodelovávaný samotný produkt linky a některé prvky použité při tvorbě linky. V TPS jsou dva druhy objektů.

- **Device** - Jsou to prvky, které nejsou částmi vyráběného produktu, ale mohou to být objekty, které se podílí na výrobě. (stůl, chapadlo, robot, a další).
- **Part** - Prvky, které jsou částmi vyráběného produktu (materiál). U těchto prvků je nutné vytvářet jejich tok.

Na obrázku 6 lze vidět složenou boudu, která se skládá z 5ti částí a základny. Na obrázku 5 jsou zobrazeny jednotlivé části boudy. Bouda se skládá z těchto tří stěn a dvou stejných částí střechy. Bloky jsou složeny z jednotlivých LEGO kostek, z důvodu ušetření počtu robotických operací. Složené bloky byly vytvořené v Invertoru jako soustava a poté vyexportovaný do .jt formátu. Bouda se skládá na základnu, která je zobrazena na obrázku 5. Z pohledu TPS jsou tyto bloky brány jako *part*, aby bylo možno u nich nastavovat generování a jejich tok.

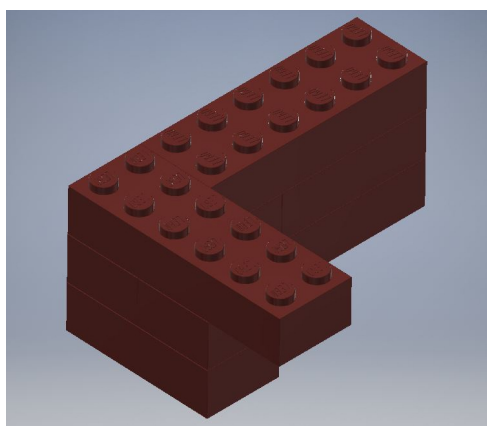
## 2.3 Generování a život produktu

Prvním krokem pro nastavení generování materiálu je vytvoření trajektorií generování pro vytvořený materiál (*part*). Trajektorie nastaví směr odkud kam se má generovaná část pohybovat. Trajektorie jsou vytvořené pro každý *part* (levou, pravou, zadní stěnu a pro dva střešní díly). Generování potřebných částí pro sestavení boudy je řízeno pomocí TPS v logických blocích. Na obrázku 8 lze vidět nastavení pro generování zadní stěny boudy. Vstupem pro logický blok je senzor, který snímá dojezdovou pozici vygenerované části. Výstupem logického bloku je zavolání operace pro vygenerování materiálu. Uvnitř logického bloku obrázek 8 je napsaná podmínka pro vygenerování. Pro vygenerování komponenty se nesmí nacházet vygenerovaná komponenta na snímané pozici. Obrázek 9 zobrazuje vývojový diagram pro vygenerování komponenty. Vygenerovaný materiál na dojezdové pozici je na obrázku 7.

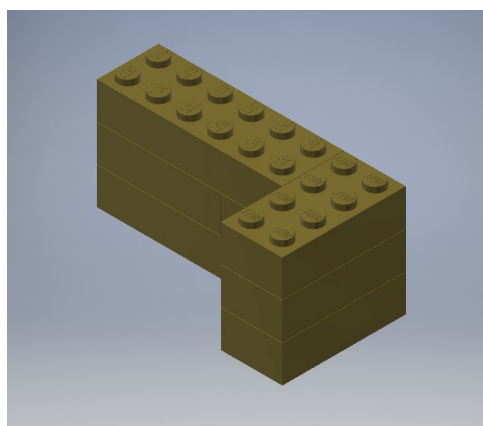
Pro takto vygenerovanou komponentu se musí nastavit její tok. Tento tok zajistí, že vygenerovaný part hned po vygenerování nezmizí a musí se provést veškeré operace, ke kterým je komponenta připojena. Každá vygenerovaná komponenta v simulaci musí projít robotickými operacemi R4, R1 nebo R2 a R3, na konci dopravníku pro odevzdání hotového výrobku (*conv\_P*) je vytvořená logika pro ukončení materiálového toku. Jakmile výrobek dojde na snímanou pozici senzoru *x\_Conv\_P\_bouda*, zastaví se dopravník *Conv\_P* a po deseti vteřinách celý výrobek zmizí, ukončení je vytvořené v logickém bloku. Na obrázku 10 je zobrazený materiálový tok, který se nachází přímo v TPS, na obrázku lze vidět všechny operace, přes které musí materiál projít. Rozmístění všech dopravníků je na obrázku 12 a senzorů na obrázku 22.

## 2.4 Dopravníky

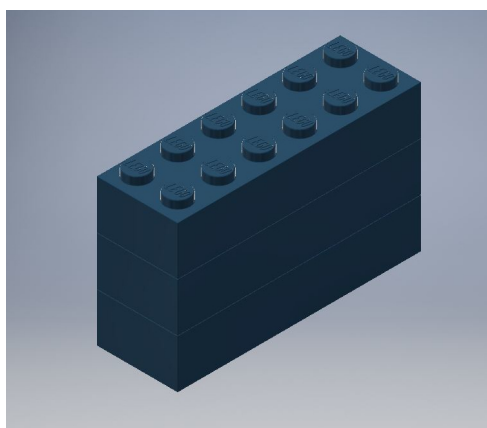
Pomocí dopravníků je zajištěn pohyb budovaného výrobku k pracovištím. Dopravník je oválného tvaru s šesti odbočkami k pracovištím a jedním koncovým dopravníkem, na kterém je výsledný produkt vyzvednut. Po těchto dopravnících se pohybuje paletka, u které lze nastavit její nosnost. Pro vytvoření dopravníku je potřeba vytvořit křivku, která bude odpovídat tvaru a délce daného dopravníku. V nástroji pro definování dopravníku se zvolí vytvořená křivka, tím se na-definuje dopravník. Pro dopravník lze nastavit jeho maximální rychlost, tolerance pro velikost přepravovaného objektu a toleranci kolizí pro tvoření fronty (*Collision tolerance*). Toleranci kolizí se nastaví vzdálenost mezi objekty při vytvoření fronty. Nadefinovaný dopravník lze vidět



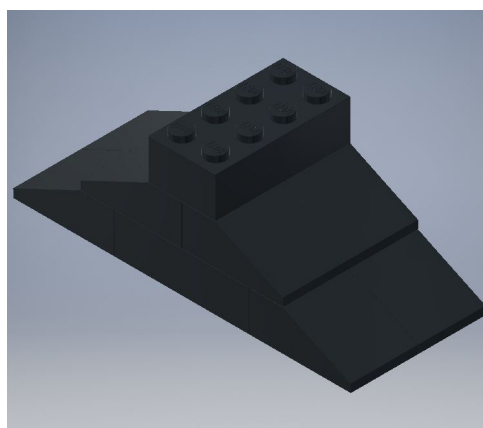
(a) Levá stěna



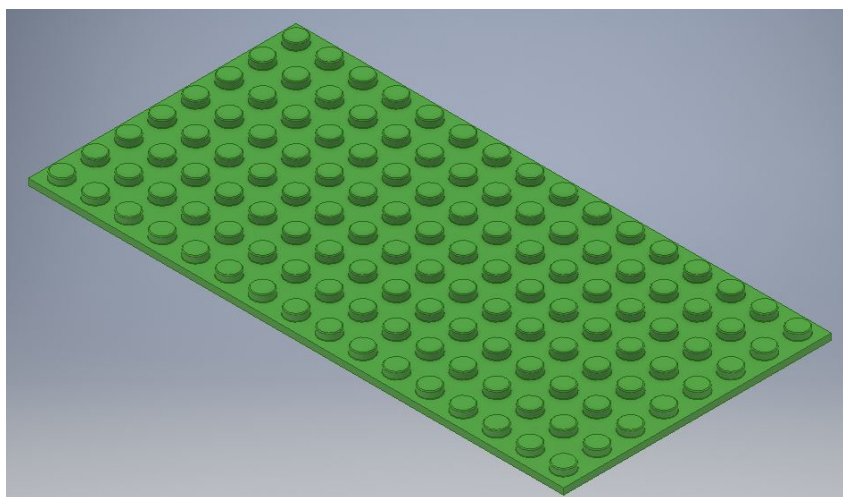
(b) Pravá stěna



(c) Zadní stěna

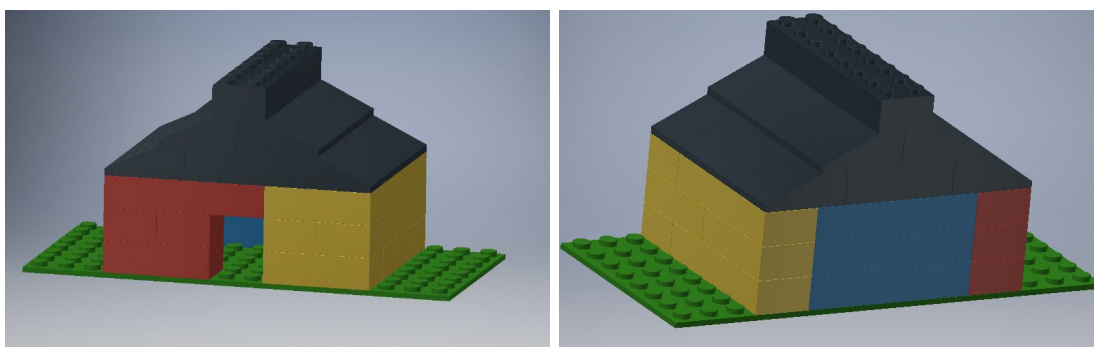


(d) Střecha



(e) Základna

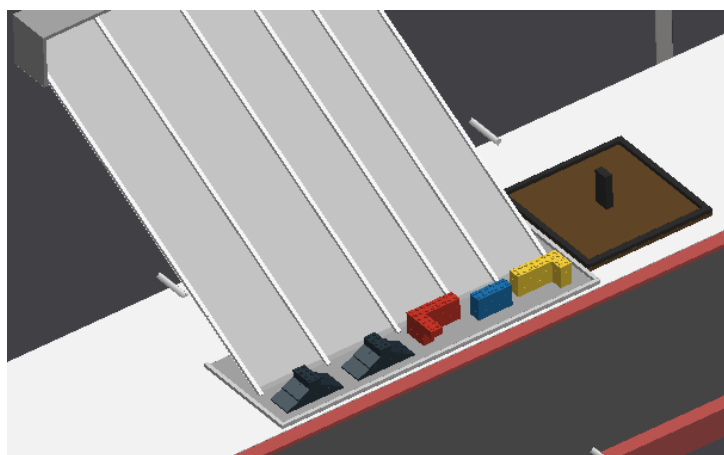
Obrázek 5: Části boudy



(a) Přední pohled

(b) Zadní pohled

Obrázek 6: Složená bouda



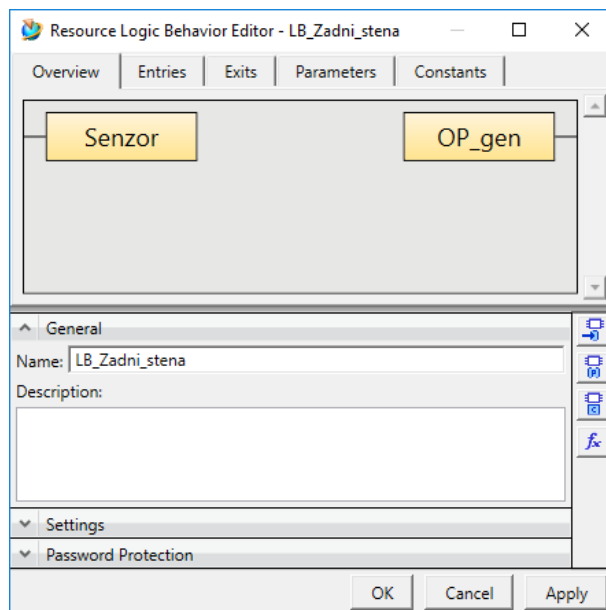
Obrázek 7: Vygenerovaný materiál

na obrázku 11. Linka obsahuje 13 jednosměrných a 7 obousměrných. Jednotlivé názvy a směry dopravníků jsou zobrazeny na obrázku 12.

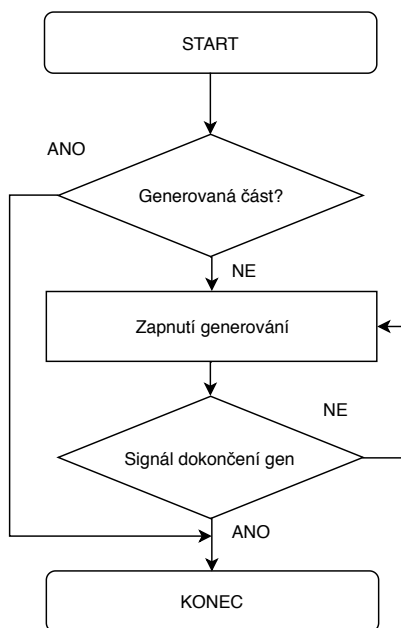
Z pohledu řízení těchto dopravníků se musí vytvořit signály pro řízení. Pro každý dopravník lze vytvořit tyto signály:

- **Start** - Zapíná dopravník
- **Stop** - Vypíná dopravník
- **Change Direction** - Pro vyvolání změny směru dopravníku
- **Change Speed** - Pro změnu rychlosti dopravníku

Signály *Start* a *Stop* jsou vytvořené u všech dopravníků. Signál pro změnu směru *ChangeDirection* je vytvořen pouze pro obousměrné dopravníky. Signál pro změnu rychlosti nebyl využit u žádného dopravníku. Pro zapnutí dopravníku musí být na signálu *Start* log.1 a na signálu *Stop* log.0. Pro zastavení dopravníku musí být signály přesně naopak, log.1 na signálu *Stop* a log.0



Obrázek 8: Ukázka logického bloku

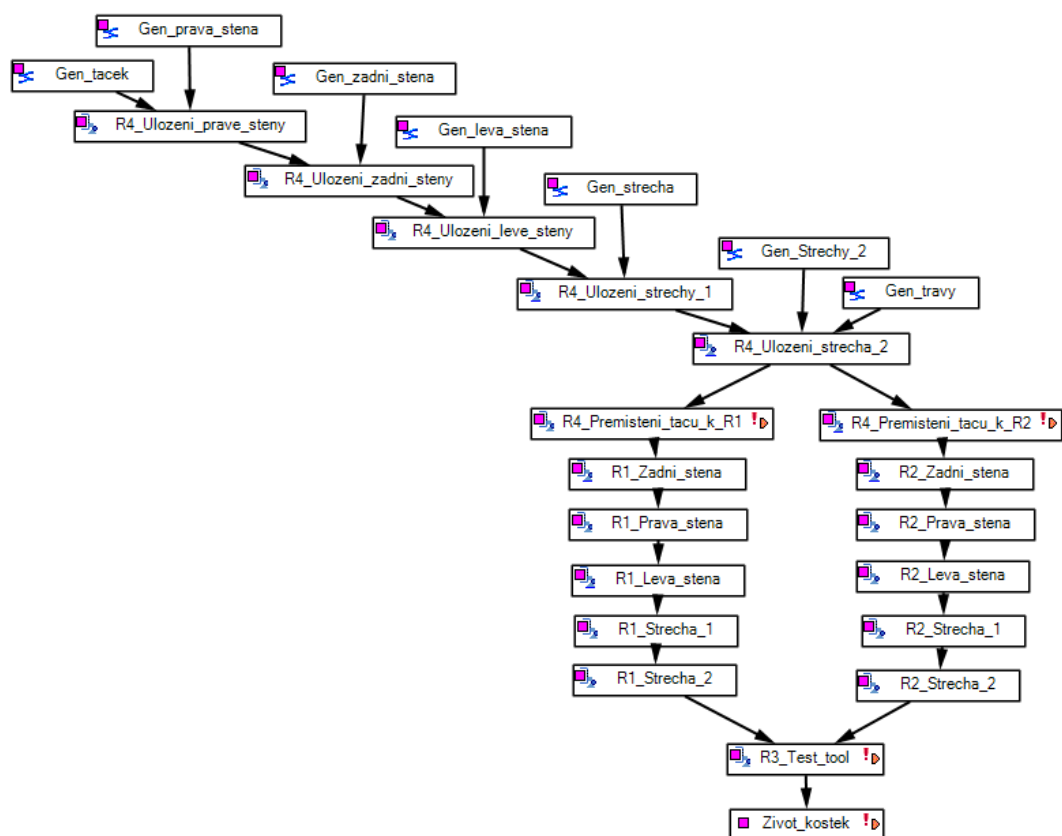


Obrázek 9: Vývojový diagram generování materiálů

na signálu *Start*. Změna směru dopravníku se provádí přivedením pulzu na signál *ChangeDirection*. TPS neposílá žádný signál o tom, jakým směrem se dopravník v danou chvíli pohybuje.

## 2.5 Robotická ramena a operace

Robotická ramena na pracovištích R1 - R3 byla zvolena tak, aby měla dostatečný dosah na stůl, kde se bude nacházet ták s připraveným materiálem pro práci, nebo testovací náradí. Použité



Obrázek 10: Tok materiálu

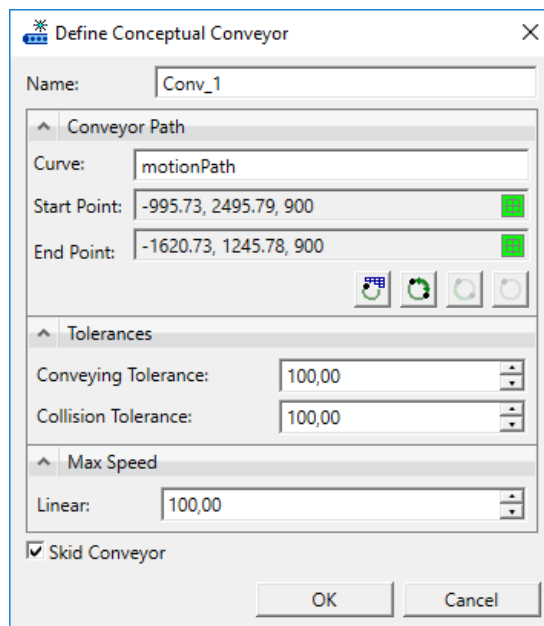
robotické rameno lze vidět na obrázku 17. Robotické rameno R4 je poměrně větší než ostatní, je umístěno na pojezdu, který umožňuje přepravu naloženého materiálu na stůl k R1 a R2.

Pro práci s objekty je potřeba k robotickému rameni upevnit chapadlo. Chapadlo lze v TPS upevnit pomocí nástroje *mount tool*. Pro vybrané chapadlo je potřeba nastavit pozice pro čelisti. Na obrázku 13 (a) je editor, ve kterém se vytvářejí a upravují pozice čelistí. Na obrázku 13 (b) je otevřena pozice *OPEN*, která ukazuje dva prsty chapadla ( $j1$ ,  $j2$ ) a jejich pozice při rozevření, tuto pozici lze upravovat dle potřeby. Aby byl přenášený objekt zvednut musí se prsty dotýkat stěn objektu, který je potřeba uchytit. Vytvořené pozice:

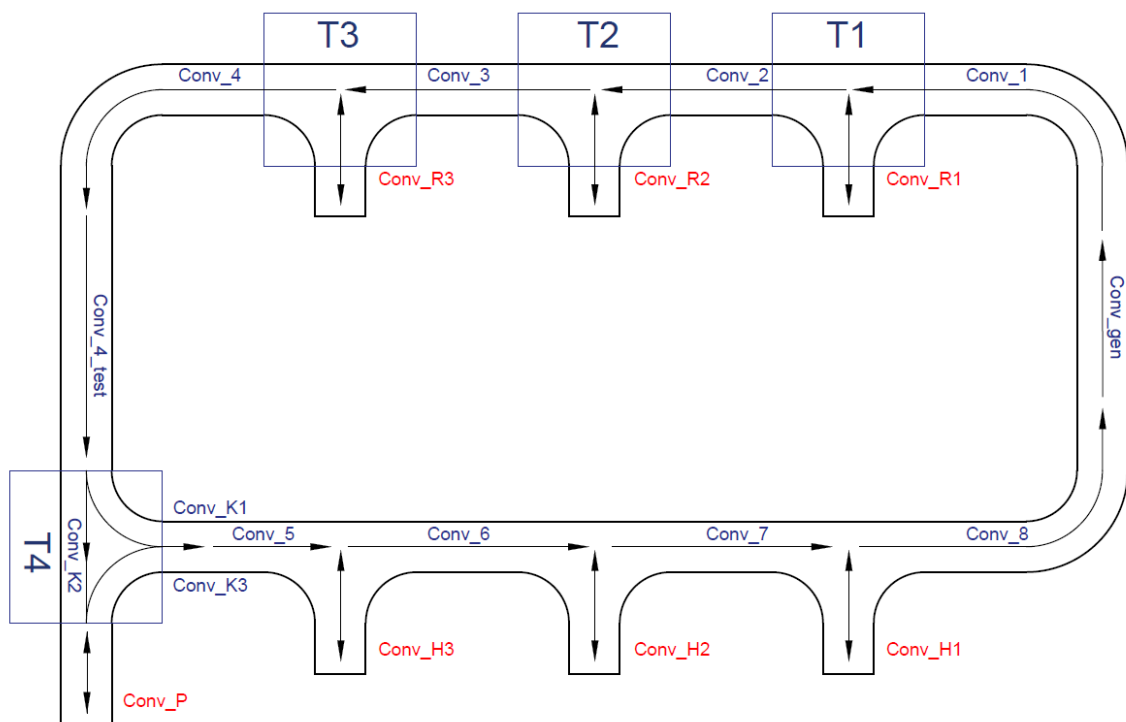
- OPEN - Pozice pro uvolnění uchopeného objektu, zároveň výchozí pozice.
- CLOSE - Pozice uchycení jednotlivých částí boudy.
- CLOSE\_2 - Pozice uchycení testovacího nástroje.

Všechny robotické operace se skládají z Pick and Place operací. Při tvorbě Pick and Place operace musí být vybrané robotické rameno, pro které je tvořena robotická operace. K robotickému ramenu musí být připevněno chapadlo. Při tvorbě trajektorií je potřeba zvolit vytvořené



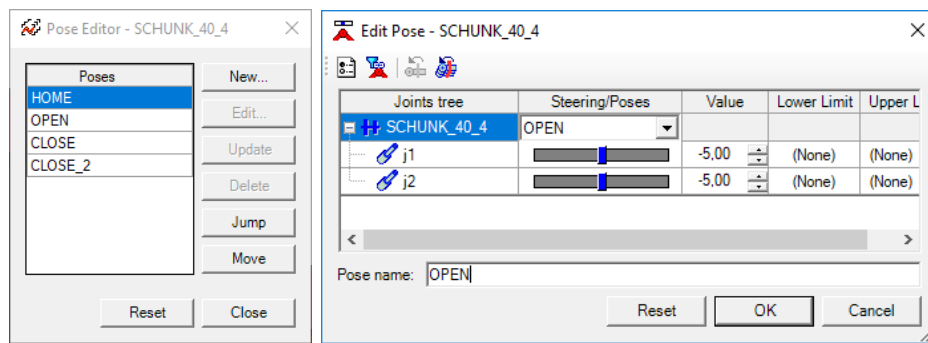


Obrázek 11: Nastavení dopravníku



Obrázek 12: Směry a názvy dopravníků

pozice v chapadlu pro uchycení a uvolnění přenášeného objektu a zvolit pozice a místo vyzvednutí a položení přenášeného objektu (*pick6* a *place6* obrázek 14). Lze přidávat další pozice, kterými musí robotické rameno projít. Jednotlivé operace lze upravit v osách x,y,y a upravo-



(a) Pose editor

(b) Nastavení a úprava pozic

Obrázek 13: Výpis pozic

vat také natočení v Rx,Ry,Rz. Na obrázku 14 je zobrazena robotická operace pro rameno R1, které umisťuje modrou stěnu boudy, celkový počet pozic, kterými musí rameno projít je sedm. Robotická operace je vložena v *Patch editoru*.

R1_Zadni...								9.53		R1
via33		-1155.72	1505.42	1050.00	0.00	0.00	0.00	1.82		
pick6		-1155.72	1505.42	925.00	0.00	0.00	0.00	0.89	# Destination	
via34		-1155.72	1505.42	1050.00	0.00	0.00	0.00	0.73		
via35		-1116.68	1245.80	1050.00	0.00	0.00	90.00	2.05		
place6		-1116.68	1245.80	931.40	0.00	0.00	90.00	1.06	# Release	
via36		-1116.68	1245.80	1050.00	0.00	0.00	90.00	0.91		
via37		-1162.19	1071.39	1224.65	-0.00	0.00	0.00	2.06		

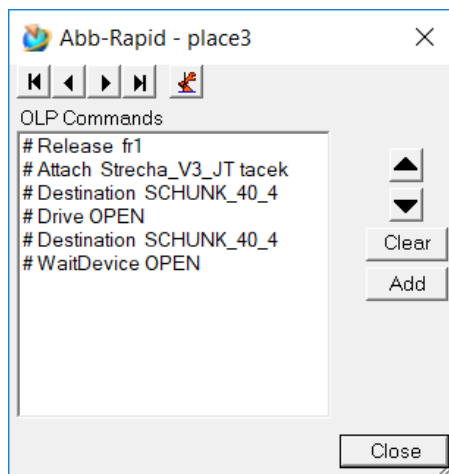
Obrázek 14: Robotická operace

Pro jednotlivé pozice robotických operací lze nastavit mnoho možností (např. změnu barvy objektu, zmizení objektu, připevnění k jinému objektu, rozpojení objektů a další). Nastavení se vytváří ve sloupečku *OLP command* v *Patch editoru*, který je zobrazený na obrázku 14. Obrázek 15 zobrazuje veškeré OLP příkazy. Příkaz *#Attach\_Strecha\_V3* byl vytvořen, zbytek příkazů vygeneroval TPS. Vygenerovaný příkaz spojí během pokládání dva objekty do celku. V tomto případě podle obrázku 16 se spojí střešní díl s táčem. Při odebrání materiálu z tácu jsou spojené objekty rozpojeny.

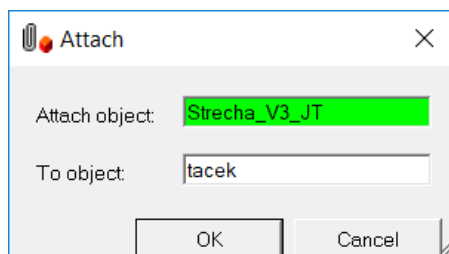
U každého robotického ramene lze vytvořit osm základních signálů. Další vytvořené signály pracují stejně jako signál *HOME*, kdy signalizují pozici ramene, pro kterou byl signál vytvořen. Například vytvoření pozice pro servis robotického ramene. Tuto pozici lze vytvořit pro rameno a poté pro ni vytvořit signál, který vyše log.1, když se robotické rameno dostane do této polohy. Výpis všech robotických signálů se nachází v tabulce 1.

### 2.5.1 Robotická pracoviště pro skládání výrobků

Robotická ramena R1 a R2, mají vytvořené robotické operace pro skládání výrobku, robotické rameno použité pro skládání lze vidět na obrázku 17. Každá část výrobku má svou robotickou operaci. Robotické operace se skládají z pozic pro vyzvednutí, umístění a dalších pozic, které



Obrázek 15: OPL menu

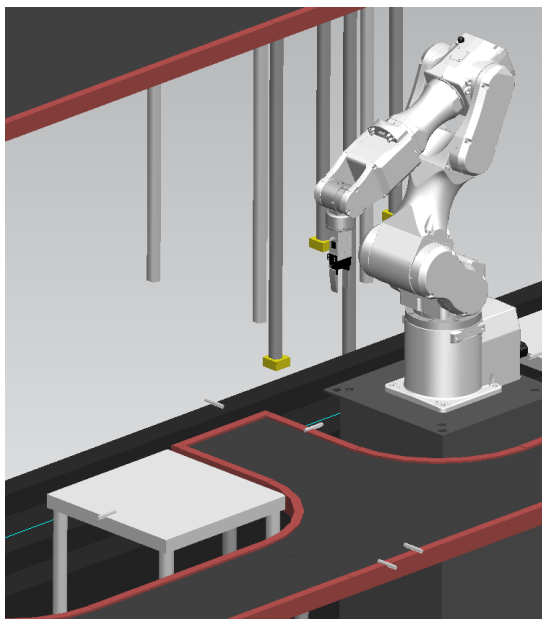


Obrázek 16: Spojování objektů pomocí operací

Tabulka 1: Tabulka robotických signálů

Název	Typ	I/O	Popis
startProgram	BOOL	Q	Spustí robotickou operaci
programNumber	BYTE	Q	Programové číslo operace, kterou má robotické rameno vykonat
emergencyStop	BOOL	Q	Záchranné zastavení robota
programPause	BOOL	Q	Pozastaví prováděnou robotickou operaci
progrmEnded	BOOL	I	Signalizuje, že robotické rameno dokončilo operaci
mirrorProgramNumber	BYTE	I	Posílá programové číslo, které bylo nahrané do robotického ramene
errorProgramNumber	BOOL	I	Signalizuje špatně nahrané číslo
robotReady	BOOL	I	Signalizuje, jestli je robotické rameno připraveno k práci
HOME	BOOL	I	Signalizuje, jestli je robot v pozici HOME

zajistí, aby se přenášený objekt a rameno s ničím nesrazily. Robotické operace se nejen odlišují svými trajektoriemi, ale i v operacích během ukládání a vyzvedávání částí boudy. V operaci pro uchycení části boudy je nastaveno odpojení části od tácu, protože jsou s ním během přepravy spojeny. V operaci pro uložení je nastaveno spojení jednotlivých částí k jiným. Bloky stěn se spojí s základnou. Střešní části se upevní ke stěně, na které jsou umístěny. K jednotlivým robotickým operacím jsou přiřazeny čísla, pomocí kterých se volají jednotlivé robotické operace. Volání těchto operací je provedeno pomocí signálu *programNumber*. V tabulce 2 jsou vypsány veškeré robotické operace a k nim přiřazená programová čísla.



Obrázek 17: Robotické rameno R1

Tabulka 2: Čísla robotických operací pro R1 a R2

Název	Číslo operace	Popis
R1_zadni_stena	10	Umístí zadní stěnu boudy na základnu
R1_prava_stena	11	Umístí pravou stěnu boudy na základnu
R1_leva_stena	12	Umístí levou stěnu boudy na základnu
R1_strecha1	13	Umístí první část střechy
R1_strecha2	14	Umístí druhou část střechy

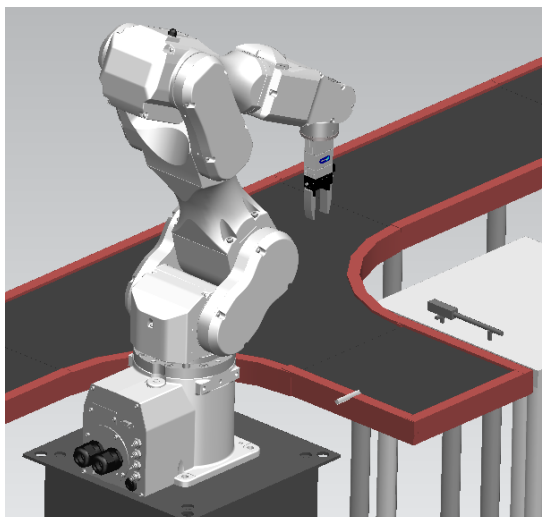
Robotická čísla pro R2 jsou přiřazené úplně stejně jak u R1, změna je pouze v názvu operací, kde místo R1 je R2. Robotické operace mají stejné čísla z důvodu usnadnění tvorby řídicích algoritmů pro jednotlivé ramena.

### 2.5.2 Robotické pracoviště pro testování

Testovací robotické stanoviště R3 má na pomocném stolku umístěný pomocný nástroj, který uchopí a provede testování. Robotické rameno má jednu robotickou operaci, která je vypsána v tabulce 3. Na obrázku 18 lze vidět pracoviště spolu s nástrojem, který je umístěný na stole.

Tabulka 3: Čísla robotických operací pro R3

Název	Číslo operace	Popis
R3_Test	30	Provede testování



Obrázek 18: Robotické rameno R3

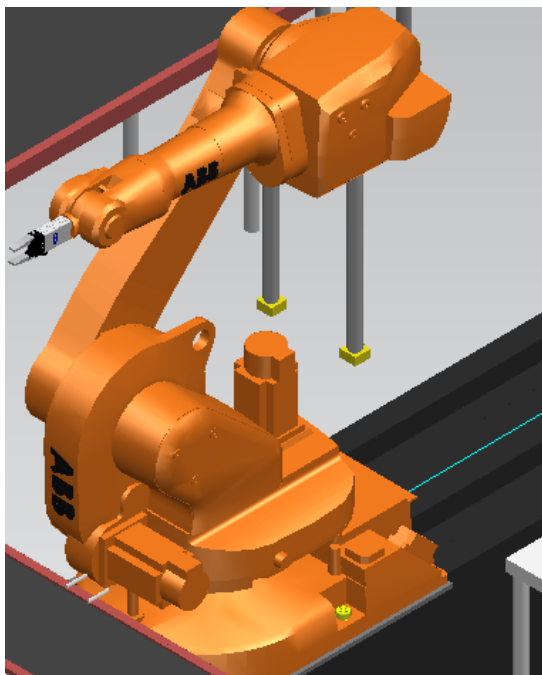
### 2.5.3 Robotické rameno R4

Robotické rameno R4 je postaveno na pojezdu, zobrazeno na obrázku 19. Tento pojezd je v TPS vytvořen jako externí osa, po které se může robot přemísťovat. V kinematic editoru je vytvořen pohyb jednotlivých kloubů u robotického ramene. Barevné čtverce (linky) představují konstrukci jednotlivých částí ramene. Jednotlivé čtverce (linky) jsou propojeny pomocí šipky (joint), ve kterých lze nastavit druh pohybu, rychlost a zrychlení osy. Lze nastavit dva druhy pohybu.

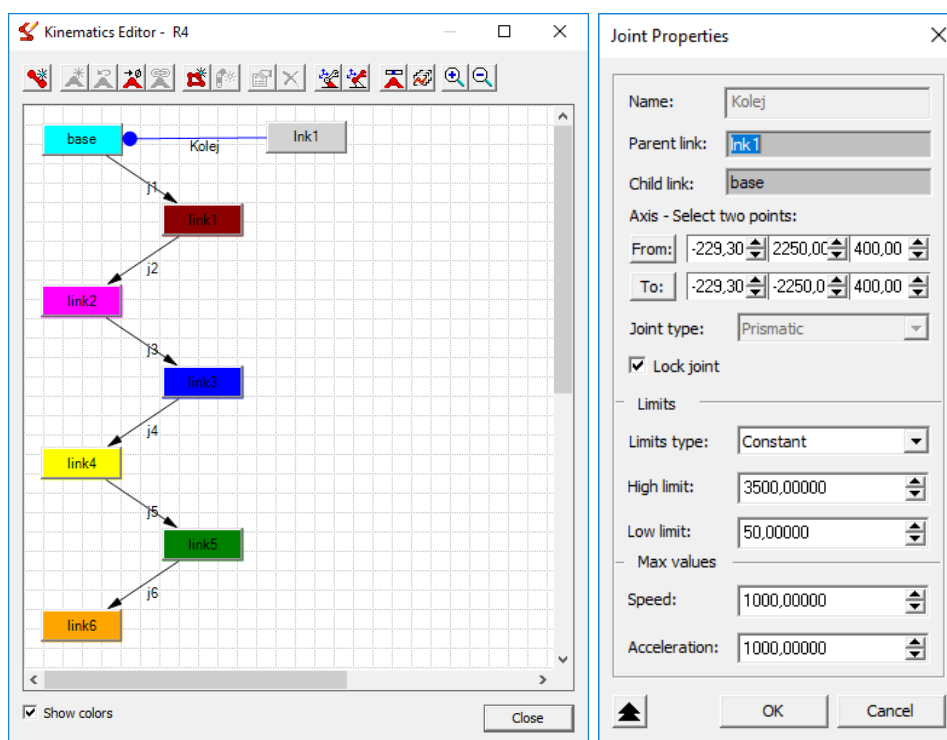
- **Rotační** - Otáčí se kolem zadané osy
- **Lineární** - Pohybuje se podél zadané osy

Na obrázku 20 (a) lze vidět jednotlivé osy (j1 - j6) a externí, která je nazvána *Kolej*, ta je připojena k základně robota. Nastavení externí osy je zobrazeno na obrázku 20 (b). Osa je zvolena jako lineární. Pro zvolení bodů (od, do) je na podstavě pojezdu vytvořená křivka, kde konce křivky jsou začátek a konec pojezdu. Lze nastavit limitní hodnoty pro pohyb na ose.

Vygenerovaný materiál rameno naskládá na ták, materiál s ním při ukládání spojí, poté převez k jednomu z pracovišť R1 nebo R2. Pro každou pozici v robotické operaci se nastaví pozice na externí ose. Pojezd nastavenou rychlostí poté jezdí mezi jednotlivými pozicemi. Například pro nabírání materiálu z blistrů je pozice na externí ose 50 a pro odložení tácu k pracovišti R1 je pozice 1500. Veškerá čísla operací, které má robotické rameno uloženo, jsou vypsána v tabulce 4.



Obrázek 19: Robotické rameno R4



(a) Kinematic editor

(b) Parametry osy

Obrázek 20: Vytváření a parametrování externí osy

Tabulka 4: Čísla robotických operací pro R4

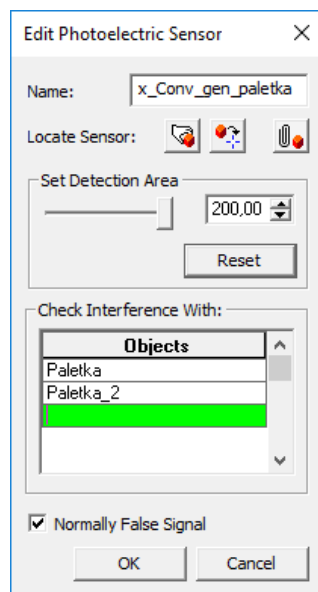
Název	Číslo operace	Popis
R4_Levá_stěna	40	Umístění levé stěny na táč
R4_Zadní_stěna	41	Umístění zadní stěny na táč
R4_Zadní_stěna	42	Umístění pravé stěny na táč
R4_Střecha1	43	Umístění první části střechy na táč
R4_Střecha2	44	Umístění druhé části střechy na táč
R4_k_R1	45	Přemístění tácu k robotickému pracovišti R1
R4_k_R2	46	Přemístění tácu k robotickému pracovišti R2

## 2.6 Senzory

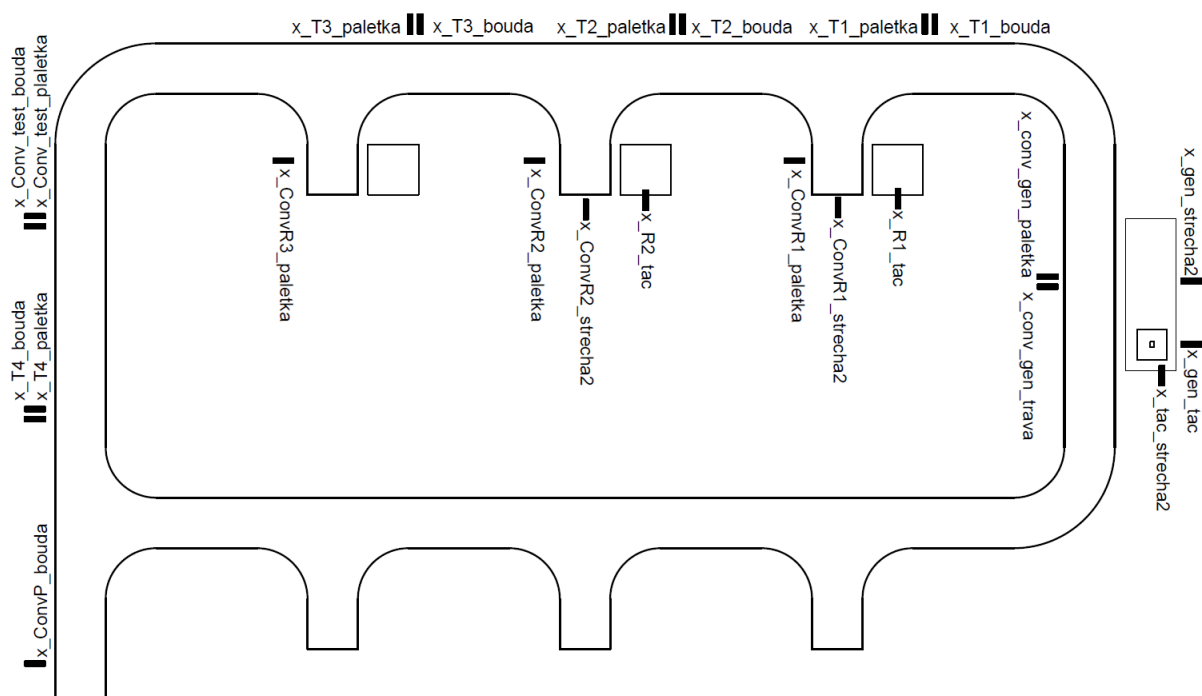
TPS má na výběr z 5ti druhů senzorů.

- **Joint Distance Sensor** - Senzor monitorující hodnotu snímaného kloubu.
- **Joint Value Sensor** - Senzor, který monitoruje dosažení nastavené hodnoty u kloubu. Možnost volby druhu signálu na výstupu jako je pulz, náběžná a sestupná hrana.
- **Proximity Sensor** - Senzor snímající vzdálenost od snímaného objektu. Jestliže objekt překročí nastavenou vzdálenost, senzor vyšle signál. Lze snímat více objektů.
- **Photoelectric Sensor** - Senzor, který vysílá paprsek o nastavené délce. V senzoru lze nastavit objekty, které má snímat. Jakmile tyto objekty přetnou paprsek, senzor vyšle signál.
- **Property Sensor** - Senzor, který detekuje změna objektů. Změna, kterou má detekovat, se musí nastavit.

V simulaci byl použit pouze *Photoelectric sensor*. Nastavení jednoho senzoru lze vidět na obrázku 21. V parametrech senzoru lze přidat objekty, které má snímat a vzdálenost dosahu světelného paprsku. Senzory jsou použity pro snímání generování materiálu, snímání pozice paletky a výrobku na dopravnících. Veškeré senzory jsou v návrhu pojmenovány *x\_místo kde se senzor nachází\_snímaný objekt*. Na obrázku 22 lze vidět rozmístění jednotlivých senzorů.



Obrázek 21: Nastavení senzorů



Obrázek 22: Rozmístění senzorů



### 3 Funkční analýza

Funkční analýza je provedena z důvodu pochopení, jak má linka fungovat a popis by měl usnadnit tvorbu řídicích algoritmů. Výstupem funkční analýzy budou tabulky se vstupními a výstupními signály. Funkční analýza je rozdělená do více částí, které zabývají problematikou jednotlivých úseků linky. V TPS je signálová logika brána z pohledu PLC. Generování materiálů je řízené pomocí TPS, proto nebude tvořeno, vytvořené řízení je popsáno v kapitole 2.3.

#### 3.1 Dopravníky

V této kapitole budou dopravníky rozděleny do čtyř částí. První částí je dopravník, na který se generuje základna po přijetí paletky. Druhou částí jsou křižovatky, které se nacházejí u pracovišť R1 - R3. Třetí část textu se zabývá druhým testováním. Poslední část se zabývá poslední křižovatkou. Veškeré dopravníky a křižovatky popsány v této kapitole jsou vykresleny na obrázku 12. Rozmístění senzoru je na obrázku 22, výpis všech signálů ze senzorů je v tabulce 8.

##### 3.1.1 Dopravník pro generování základny

Dopravník *Conv\_gen* je potřeba zastavit, jakmile se na paletce nenachází základna a je na pozici senzoru *x\_conv\_gen\_paletka*. Jakmile jsou splněny obě podmínky, spustí se operace pro generování zelené základny, která je řízená pomocí TPS. Dopravník se spustí, jakmile senzor *x\_conv\_gen\_trava* zachytí základnu.

##### 3.1.2 Podmínky pro řízení dopravníků u robotických pracovišť

Aby mohla paletka zajet k pracovištím R1 - R3, musí splnit podmínky pro zajetí. Na začátku každé křižovatky se nachází dvojice senzorů, jeden senzor snímá paletku, druhý výrobek. U pracovišť se nachází senzory, které snímají obsazenost pracovišť.

Jakmile se dostane paletka na pozici první křižovatky (T1), senzor pro snímání paletky *x\_T1\_paletka* a boudy *x\_T1\_bouda* vyhodnotí, zda převáží nebo nepřeváží boudu. Když boudu nepřeváží a není spuštěn dopravník *Conv\_R1* a pracoviště R1 je volné, může být paletka poslána na dopravník *Conv\_R1* k pracovišti R1. Když je paletka posílána na dopravník *Conv\_R1*, musí se vypnout dopravník *Conv\_2*, jinak by paletka jela dál po oválné trase. Dopravník *Conv\_2* lze spustit, jakmile paletka dojde na pozici senzoru *x\_Conv\_R1\_paletka*. Na této pozici se musí vypnout dopravník *Conv\_R1* a změnit směr pohybu. Během vyjždění nebo zajždění paletky k R1 je potřeba hlídat příjezd paletky po dopravníku *Conv\_1*, jakmile by se dostala paletka na pozici senzoru *x\_T1\_paletka*, je potřeba vypnout dopravník *Conv\_1*, nebo jej vypnout rovnou při přepravě paletky od nebo k pracovišti. Vyjždění paletky se nesmí spustit, když se přes křižovátku pohybuje druhá paletka. Po vyjetí paletky z dopravníku *Conv\_R1* se musí tento dopravník vypnout a změnit jeho směr. Jakmile je pracoviště R1 obsazené a přijede druhá paletka, je tato paletka vyslána k druhému pracovišti R2. Během přejezdu přes křižovátku

se nesmí spustit dopravník *Conv\_R1*, i když by se na paletce zrovna složil výrobek, musí počkat na pracovišti. Kdyby paletka na vstupu křižovatky (T1) bouda převážela, je paletka poslána dále přes křižovatku k další křižovatce.

Na vstupu druhé křižovatky (T2) se zase nachází dvojice senzorů, které snímají paletku *x\_T2\_paletka* a boudu *x\_T2\_bouda*, které vyhodnotí, zda paletka zajede k pracovišti R2 nebo bude pokračovat dále. Jakmile paletka převáží hotový výrobek, není spuštěn dopravník *Conv\_R2*, může paletka pokračovat dále přes křižovatku k otestování na pracovišti R3. Jestli nepřeváží hotový výrobek, je potřeba poslat paletku k pracovišti R2. Podmínky pro spuštění dopravníku *Conv\_R2* jsou stejné jak u první křižovatky (T1).

Složený výrobek musí být otestován pomocí robotického ramene R3. Na vstupu třetí křižovatky (T3) se nachází dvojice senzorů pro snímání paletky *x\_T3\_paletka* a boudy *x\_T3\_bouda*. Když na tuto křižovatku přijede paletka měla by vždy převážet hotový výrobek. Při splnění podmínek volného pracoviště R3 a vypnutého dopravníku *Conv\_R3*, může paletka zajet k pracovišti. Při zajíždění k pracovišti je potřeba vypnout dopravník *Conv\_4*. Vypnutí dopravníku *Conv\_R3* proběhne jakmile paletka přijede na pozici senzoru *x\_Conv\_R3\_paletka*. Dopravník je spuštěn, jakmile robotické rameno R3 vyšle hodnotu *true* na signál *R3\_programEnded* a proběhla změna směru. Dopravník *Conv\_R3* je vypnut, když jej paletka opustí a musí se provést změna směru. Když je pracoviště R3 obsazeno, paletka vyčká na pozici senzoru *x\_T3\_paletka* a jakmile se uvolní místo na pracovišti a proběhne vyvezení paletky, spustí se dopravník *Conv\_3*.

### 3.1.3 Druhé testování

Na dopravníku *Conv\_4\_test* probíhá druhé testování. Na dopravníku se nachází senzor pro snímání paletky *x\_Conv\_test\_paletka* a boudy *x\_Conv\_test\_bouda*. Když na tyto senzory přijede paletka se složenou boudou, dopravník *Conv\_4\_test* se zastaví. Testování je pouze simulační. Až se provede testování, dopravník *Conv\_4\_test* se spustí.

### 3.1.4 Křižovatka pro odevzdání výrobku

Na začátku křižovatky pro odevzdání materiálu (T4) se nacházejí senzory pro snímání paletky *x\_T4\_paletka* a boudy *x\_T4\_bouda*. Při průjezdu prázdné palety se musí vypnout dopravníky *Conv\_K2* a zapnout dopravník *Conv\_K1*, prázdná paletka se nesmí dostat na dopravník *Conv\_P*. Po průjezdu naložené paletky může paletka pokračovat k dopravníku *Conv\_P*, jestliže je dopravník *Conv\_P* volný a nepohybuje se směrem k a od dopravníku paletka, když ano musí se paletka zastavit na pozici senzorů a počkat, až se uvolní dopravník *Conv\_P*. Když je křižovatka volná, může paletka po dopravníku *Conv\_K2* pokračovat k dopravníku *Conv\_P*. Dopravníky *Conv\_K1* a *Conv\_K3* se musí vypnout, když je spuštěný dopravník *Conv\_K2*. Jakmile se paletka nachází na dopravníku *Conv\_P* a dojde na senzor *ConvP\_bouda*, musí se vypnout dopravník *Conv\_P*. Když je bouda na pozici senzoru *ConvP\_bouda* a bouda zmizí, musí se změnit směr a zapnout dopravník *Conv\_P* a poslat prázdnou paletku zpět na oválný do-

pravník. Zmizení boudy (ukončení toku materiálu) je popsáno v kapitole 2.3. Když paletka jede zpět na oválný dopravník, musí být vypnut dopravník *Conv\_K2* a zapnut dopravník *Conv\_K3*.

Všechny výstupní signály pro řízení dopravníků jsou vypsány v tabulce 5. Vstupní signály nebyly použity žádné.

## 3.2 Robotická ramena

V této kapitole budou popsána veškerá robotická ramena. Jak již bylo zmíněno v kapitole 2.5 o robotických operacích, na lince se nacházejí čtyři robotická ramena. Robotická ramena R1 a R2 skládají výrobek. Rameno R3 testuje složený výrobek. Rameno R4 je umístěné na pojezdu nakládá a přepravuje materiál k pracovištím R1 a R2.

### 3.2.1 Robotické rameno R4

Aby začalo robotické rameno R4 nakládat materiál, musí být vygenerovaný materiál a tác. Táč je snímán pomocí senzoru *x\_gen\_tac* a poslední vygenerovaná kostka je snímána pomocí senzoru *x\_gen\_strecha2*. Vysláním čísla robotické operace na signál *R4\_program Number* se nastaví operace, kterou má robotické rameno vykonat. Robotické rameno toto číslo pošle zpět pomocí signálu *R4\_mirrorProgramNumber*, číslo se posílá zpět z důvodu kontroly, že robotické rameno má správné číslo. Čísla robotických operací jsou vypsány v tabulce 4. Jakmile se robotická čísla rovnají, může se spustit robotická operace. Spuštění robotické operace se provede pomocí signálu *R4\_startProgram*. Když se ukončí operace, na signálu *R4\_programEnded* bude hodnota *true*. Když se ukončí operace, může se vyslat číslo s novou operací a proces se opakuje. Robotické operace pro náklad jsou v pořadí od 40 do 44. Jakmile jsou uloženy všechny kostky, lze přesunout naložený materiál k pracovištím. Naložení tácu snímá senzor *x\_tac\_strecha2*, který snímá poslední naloženou kostku. Přesun k pracovišti se provede, když bude pracoviště bez materiálu. Materiál na pracovištích je snímán pomocí senzorů *x\_R1\_tac* pro pracoviště R1 a *x\_R2\_tac* pro pracoviště R2. Pracoviště R1 musí mít větší prioritu než R2. Operace pro přesun jsou 45 a 46. Volání operací funguje stejně jako u nakládání. Když robotické rameno odveze materiál na pracoviště, musí jej znovu naložit a poté převést na druhé pracoviště.

### 3.2.2 Robotická ramena R1 a R2

Robotická ramena R1 a R2 zahájí svoje operace pouze, pokud mají na pomocném stole táč s materiálem a připravenou paletku se základnou. Čísla robotických operací jsou vypsána v tabulce 2. Materiál na stole je snímán pomocí senzoru *x\_R1\_tac* pro pracoviště R1 a pro pracoviště R2 senzorem *x\_R2\_tac*. Paletka na pozici pro složení výrobku je snímána senzorem *x\_ConvR1\_paletka* pro pracoviště R1 a senzorem *x\_ConvR2\_paletka* pro pracoviště R2. Spuštění operací funguje stejně jako u ramene R4, pouze počet operací, které robotické rameno vykonává je jiný. Složený výrobek opustí pracoviště, když bude umístěna poslední část výrobku, která je snímána sen-

Tabulka 5: Tabulka výstupních signálů pro řízení dopravníků

Název signálu	Typ	Popis
Conv_gen_Start	BOOL	Zapíná dopravník Conv_gen
Conv_gen_Stop	BOOL	Vypíná dopravník Conv_gen
Conv_1_Start	BOOL	Zapíná dopravník Conv_1
Conv_1_Stop	BOOL	Vypíná dopravník Conv_1
Conv_2_Start	BOOL	Zapíná dopravník Conv_2
Conv_2_Stop	BOOL	Vypíná dopravník Conv_2
Conv_3_Start	BOOL	Zapíná dopravník Conv_3
Conv_3_Stop	BOOL	Vypíná dopravník Conv_3
Conv_4_Start	BOOL	Zapíná dopravník Conv_4
Conv_4_Stop	BOOL	Vypíná dopravník Conv_4
Conv_4_test_Start	BOOL	Zapíná dopravník Conv_4_test
Conv_4_test_Stop	BOOL	Vypíná dopravník Conv_4_stop
Conv_5_Start	BOOL	Zapíná dopravník Conv_5
Conv_5_Stop	BOOL	Vypíná dopravník Conv_5
Conv_6_Start	BOOL	Zapíná dopravník Conv_6
Conv_6_Stop	BOOL	Vypíná dopravník Conv_6
Conv_7_Start	BOOL	Zapíná dopravník Conv_7
Conv_7_Stop	BOOL	Vypíná dopravník Conv_7
Conv_8_Start	BOOL	Zapíná dopravník Conv_8
Conv_8_Stop	BOOL	Vypíná dopravník Conv_8
Conv_R1_Start	BOOL	Zapíná dopravník Conv_R1
Conv_R1_Stop	BOOL	Vypíná dopravník Conv_R1
Conv_R1_ChangeDirection	BOOL	Mění směr pohybu na dopravníku Conv_R1
Conv_R2_Start	BOOL	Zapíná dopravník Conv_R2
Conv_R2_Stop	BOOL	Vypíná dopravník Conv_R2
Conv_R2_ChangeDirection	BOOL	Mění směr pohybu na dopravníku Conv_R2
Conv_R3_Start	BOOL	Zapíná dopravník Conv_R3
Conv_R3_Stop	BOOL	Vypíná dopravník Conv_R3
Conv_R3_ChangeDirection	BOOL	Mění směr pohybu na dopravníku Conv_R3
Conv_K1_Start	BOOL	Zapíná dopravník Conv_K1
Conv_K1_Stop	BOOL	Vypíná dopravník Conv_K1
Conv_K2_Start	BOOL	Zapíná dopravník Conv_K2
Conv_K2_Stop	BOOL	Vypíná dopravník Conv_K2
Conv_K3_Start	BOOL	Zapíná dopravník Conv_K3
Conv_K3_Stop	BOOL	Vypíná dopravník Conv_K3

zorem  $x\_ConvR1\_strecha2$  pro pracoviště R1 a senzorem  $x\_ConvR2\_strecha2$  pro pracoviště R2.

### 3.2.3 Robotické rameno R3

Robotické rameno R3 zahájí testování, jakmile dojde paletka s hotovým výrobkem na místo, které je snímáno pomocí senzoru  $x\_ConvR3\_paletka$ . Robotické operace ramene jsou vypsány v tabulce 3. Spouštění operací probíhá stejně jako u ramene R4, které bylo popsáno v předchozí kapitole. Rameno vykoná testování a otestovaný výrobek opustí pracoviště po otestování, a to za pomoci signálu  $R3\_programEnded$ .

Veškeré signály určené pro řízení robotických ramen jsou vypsány v tabulkách. Výstupní signály jsou vypsány v tabulce 6 a vstupní signály v tabulce 7.

Tabulka 6: Výstupní signály pro řízení robotických ramen

Název	Typ	Popis
R1_startProgram	BOOL	Zapne robotické rameno, vykoná robotickou operaci dle čísla operace, které přijal
R1_programNumber	BYTE	Číslo robotické operace, kterou má robot vykonat (10-15)
R1_programPause	BOOL	Pozastaví robotickou operaci
R1_emergencyStop	BOOL	Záchranné zastavení
R2_startProgram	BOOL	Zapne robotické rameno, vykoná robotickou operaci dle čísla operace, které přijal
R2_programNumber	BYTE	Číslo robotické operace, kterou má robot vykonat (10-15)
R2_programPause	BOOL	Pozastaví robotickou operaci
R2_emergencyStop	BOOL	Záchranné zastavení
R3_startProgram	BOOL	Zapne robotické rameno, vykoná robotickou operaci dle čísla operace, které přijal
R3_programNumber	BYTE	Číslo robotické operace, kterou má robot vykonat (30)
R3_programPause	BOOL	Pozastaví robotickou operaci
R3_emergencyStop	BOOL	Záchranné zastavení
R4_startProgram	BOOL	Zapne robotické rameno, vykoná robotickou operaci dle čísla operace, které přijal
R4_programNumber	BYTE	Číslo robotické operace, kterou má robot vykonat (40-46)
R4_programPause	BOOL	Pozastaví robotickou operaci
R4_emergencyStop	BOOL	Záchranné zastavení

### 3.3 Senzory

Celkový počet senzorů, ze kterých bude čteno pomocí PLC, je 24. Zbytek senzorů je použit pro řízení přímo v TPS. Všechny vstupní signály ze senzorů jsou vypsány v tabulce 8. V tabulce se také nachází popis funkce jednotlivých senzorů.

Tabulka 7: Vstupní signály pro řízení robotických ramen

Název	Typ	Popis
R1_mirrorProgramNumber	BYTE	Číslo robotické operace, které robotické rameno pošle zpět
R1_programEnded	BOOL	Signalizuje dokončení operace
R2_mirrorProgramNumber	BYTE	Číslo robotické operace, které robotické rameno pošle zpět
R2_programEnded	BOOL	Signalizuje dokončení operace
R3_mirrorProgramNumber	BYTE	Číslo robotické operace, které robotické rameno pošle zpět
R3_programEnded	BOOL	Signalizuje dokončení operace
R4_mirrorProgramNumber	BYTE	Číslo robotické operace, které robotické rameno pošle zpět
R5_programEnded	BOOL	Signalizuje dokončení operace

Tabulka 8: Vstupní signály ze senzorů

Název	Typ	Popis
x_Conv_gen_paletka	BOOL	Detekce paletky na dopravníku Conv_gen
x_Conv_gen_trava	BOOL	Detekce vygenerované základny
x_gen_tacek	BOOL	Detekce vygenerovaného tácu
x_gen_strecha_2	BOOL	Detekce vygenerované poslední kostky
x_T1_paletka	BOOL	Detekce paletky na vstupu první křižovatky
x_T1_bouda	BOOL	Detekce výrobku na vstupu první křižovatky
x_R1_tac	BOOL	Detekce tácu na pracovišti R1
x_Conv_R1_paletka	BOOL	Detekce paletky na konci dopravníku Conv_R1
x_Conv_R1_Strecha_2	BOOL	Detekce složení výrobku na pracovišti R1
x_T2_paletka	BOOL	Detekce paletky na vstupu druhé křižovatky
x_T2_bouda	BOOL	Detekce výrobku na vstupu druhé křižovatky
x_R2_tac	BOOL	Detekce tácu na pracovišti R2
x_Conv_R2_paletka	BOOL	Detekce paletky na konci dopravníku Conv_R2
x_Conv_R2_strecha_2	BOOL	Detekce složení výrobku na pracovišti R2
x_T3_paletka	BOOL	Detekce paletky na vstupu třetí křižovatky
x_T3_bouda	BOOL	Detekce výrobku na vstupu třetí křižovatky
x_Conv_R3_paletka	BOOL	Detekce paletky na konci dopravníku Conv_R3
x_Conv_test_paletka	BOOL	Detekce paletky na dopravníku Conv_4_test
x_Conv_test_bouda	BOOL	Detekce výrobku na dopravníku Conv_4_test
x_T4_paletka	BOOL	Detekce paletky na vstupu čtvrté křižovatky
x_T4_bouda	BOOL	Detekce výrobku na vstupu čtvrté křižovatky
x_Conv_P_bouda	BOOL	Detekce výrobku na konci dopravníku

## 4 Řídicí aplikace

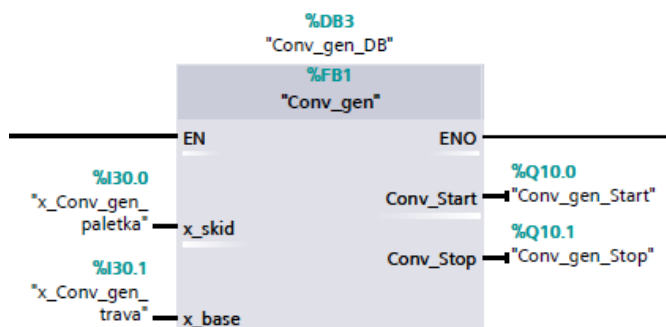
Celá aplikace je napsaná v programovacím jazyce ladder diagram (LAD) v programu TIA Portal v.14. Pro řízení a testování bylo vybráno PLC Simatic S7-1500 s CPU 1516-3 PN/DP, které se nachází ve škole. IP adresa tohoto PLC je 158.196.133.85. V jednotlivých podkapitolách jsou popsány části řídicích algoritmů, které byly navrženy pro části systému.

### 4.1 Dopravníky

Dopravníky jsou řízeny pomocí tří funkčních bloků, které je zapínají, vypínají a mění jejich směr pohybu. K rozhodování a hlídání stavů na křižovatkách jsou k tomuto účelu vytvořeny tři funkční bloky, které ze sensoriky a předchozích stavů tvoří logiku pro zapínání a vypínání jednotlivých dopravníků.

#### 4.1.1 Dopravník Conv\_gen

Funkční blok *Conv\_gen* je vytvořen speciálně pro jeden dopravník *Conv\_gen*. V tomto funkčním bloku se nachází RS obvod. Na resetovací vstup (*R*) RS obvodu je připojena podmínka pro zastavení dopravníku. Na nastavovací vstup (*S*) je připojena podmínka pro zapnutí dopravníku. Dopravník je po spuštění simulace zapnutý. Funkční blok je vidět na obrázku 23.

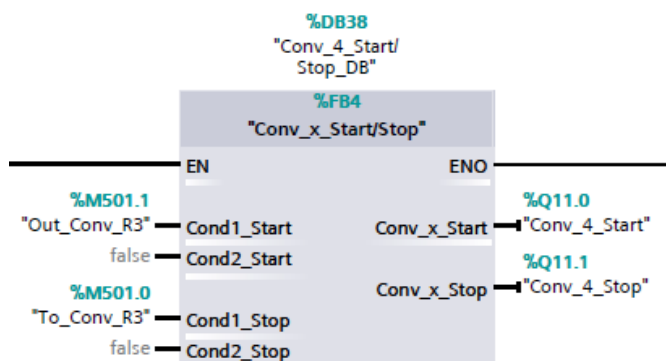


Obrázek 23: Funkční blok pro Conv\_gen

#### 4.1.2 Jednosměrné dopravníky

Funkční blok *Conv\_x\_Start/Stop* řídí jednosměrné dopravníky, až na dopravník pro generování zelené základny, ten má vlastní funkční blok. Ve funkčním bloku se nachází RS obvod, který spíná a vypíná výstup. Vstupní podmínky jsou převedeny na pulsy. Obě vstupní podmínky pro zapnutí dopravníku jsou připojeny na resetovací (*R*) vstup RS obvodu. Obě vstupní podmínky pro vypnutí dopravníku jsou připojeny k nastavovacímu (*S*) vstupu RS obvodu. Na výstup funkčního bloku se připojují výstupní signály pro zapnutí a vypnutí dopravníku. Funkční blok, který řídí

dopravník *Conv\_4* lze vidět na obrázku 24. Veškeré jednosměrné dopravníky a jejich směry jsou zobrazeny na obrázku 12.



Obrázek 24: Funkční blok pro jednosměrné dopravníky

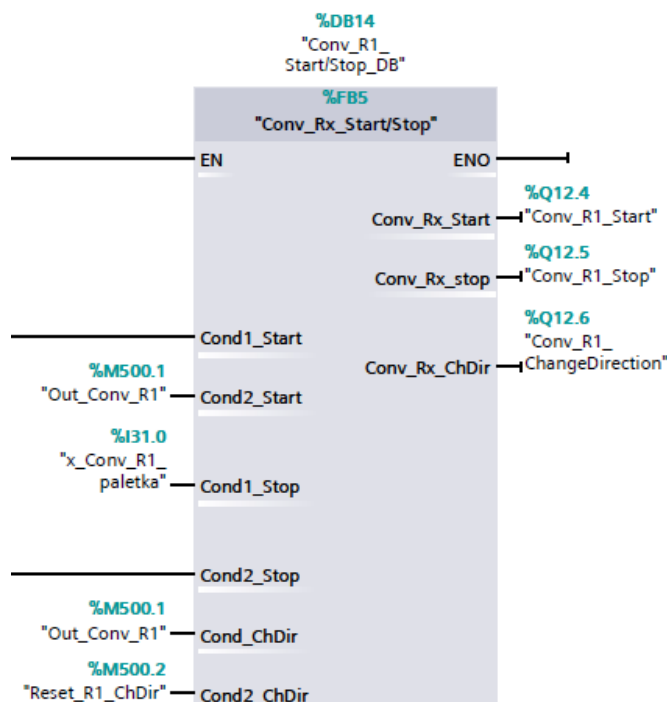
#### 4.1.3 Obousměrné dopravníky

Obousměrné dopravníky jsou řízeny pomocí funkčního bloku *ConvRx\_Start/Stop*. Dopravníky jsou po spuštění simulace vypnuty. Uvnitř funkčního bloku se nachází RS obvod, který se přepíná dle vstupních podmínek. Dvě vstupní podmínky pro zapnutí jsou připojeny k nastavovacímu (S) vstupu RS obvodu. Dvě vstupní podmínky pro vypnutí jsou připojeny k resetovacímu (R) vstupu RS obvodu. Dále má dvě vstupní podmínky pro změnu směru, které jsou připojeny na výstupní signál pro změnu směru dopravníku. Funkční blok lze vidět na obrázku 25.

#### 4.1.4 Křižovatky T1 - T3

Pohyb palet na křižovatkách T1 - T3 je zpracováván pomocí funkčního bloku *T1-T3 Control* obrázek 26. První možností řízení těchto křižovatek byla snaha vytvořit funkční blok pro každou z křižovatek, ale z důvodu dopravníků, které se nacházejí na vstupu a výstupu některých křižovatek, bylo jednodušší sloučit řízení do jednoho bloku a řešit veškeré možnosti v něm. Do tohoto funkčního bloku je připojena veškerá sensorika, která se nachází na vstupech a koncích křižovatek T1 - T3 a signály z robotických ramen R1 - R3. Výstupem z funkčního bloku jsou stavy, díky kterým se spouští a vypínají a mění směr u dopravníků, které se nacházejí na křižovatkách T1 až T3. Díky tomu lze mít nad paletkami, které se pohybují po dopravnících plnou kontrolu. Dopravníky, kterým posílá signály jsou tyto: *Conv\_1*, *Conv\_2*, *Conv\_3*, *Conv\_4*, *Conv\_R1*, *Conv\_R2*, *Conv\_R3*, dopravníky lze vidět na obrázku 12. Funkční blok má na výstupu signály *Reset\_Rx\_ChDir*, které zajistí po vyjetí paletky od pracovišť změnu směru na dopravnících *Conv\_R1*, *Conv\_R2*, *Conv\_R3*. Základní směr je k pracovišti, když se na dopravníku nenachází paletka. Hlídání směru je z důvodu, že TPS neposílá žádný signál o tom jakým směrem se paletka pohybuje, a proto je potřeba hlídat směr dopravníku.





Obrázek 25: Funkční blok pro obousměrné dopravníky

#### 4.1.5 Druhé testování

Testování v testovací stanici je řízeno funkčním blokem *Test Control*. Do funkčního bloku jsou připojeny senzory, které snímají paletku a boudu. Výstupem jsou dva stavy. Stav pro testování (*Testing\_On*) se sepne, jakmile jsou na pozici senzorů paletka s boudou. Testování se po třech sekundách zastaví. Tyto výstupy řídí dopravník *Conv\_4\_test*. Kdyby projížděla prázdná paletka, funkční blok to vyhodnotí a nespustí sekvenci testování a paletka pokračuje dál.

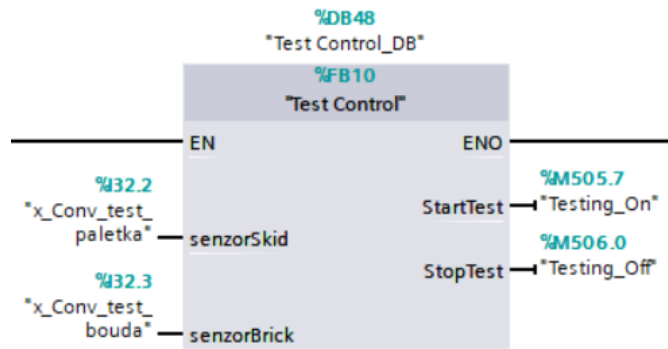
#### 4.1.6 Křižovatka T4

Poslední křižovatka T4, je řízena pomocí funkčního bloku *Control\_T4* obrázek 28. Tento blok ovládá dopravníky *conv\_K1*, *Conv\_K2*, *Conv\_K3*, *Conv\_P*. Do funkčního bloku jsou připojeny senzory (*x\_T4\_paletka*, *x\_T4\_bouda*, *x\_Conv\_P\_bouda*), které se nacházejí na vstupu křižovatky a na konci dopravníku *Conv\_P*.

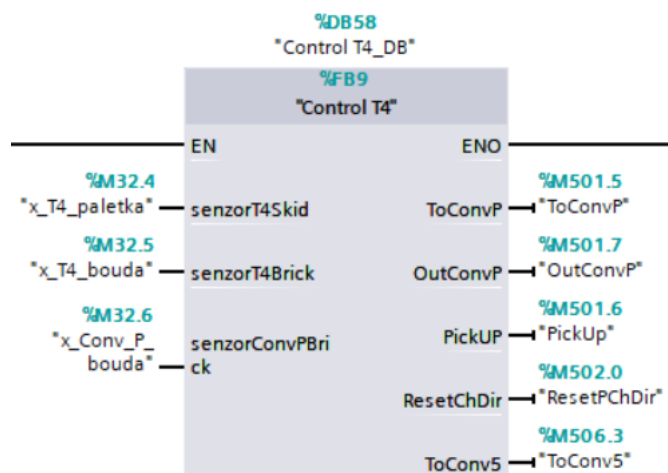
### 4.2 Robotická ramena

Robotická ramena R1 - R3 jsou řízena pomocí funkčního bloku *Rx*, zobrazený na obrázku 29. Na vstupu funkčního bloku jsou připojeny senzory pro snímání paletky, tácu na stole a uložení posledního kusu výrobku.





Obrázek 27: Funkční blok pro testování

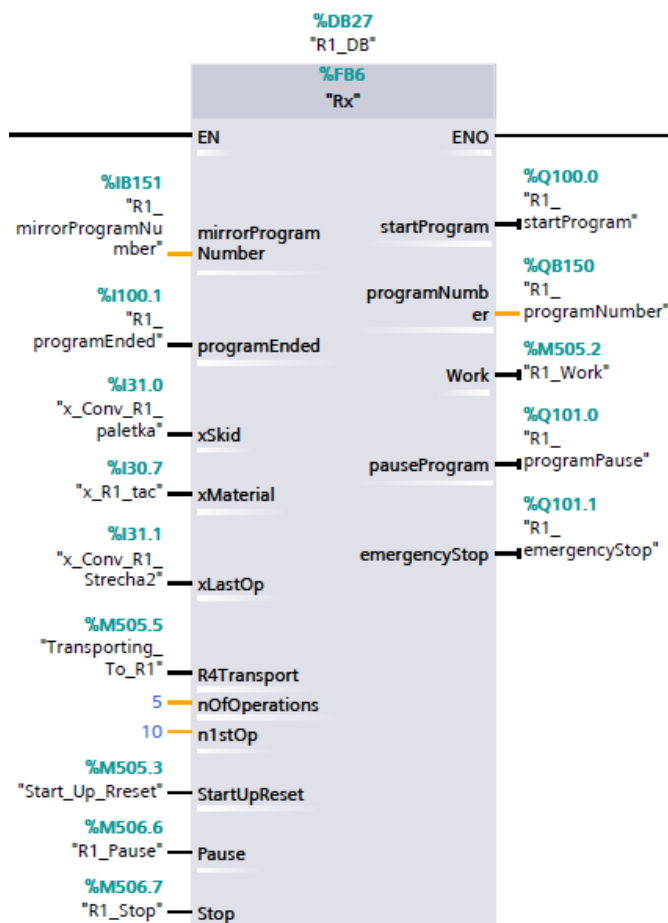


Obrázek 28: Funkční blok pro řízení křižovatky T4

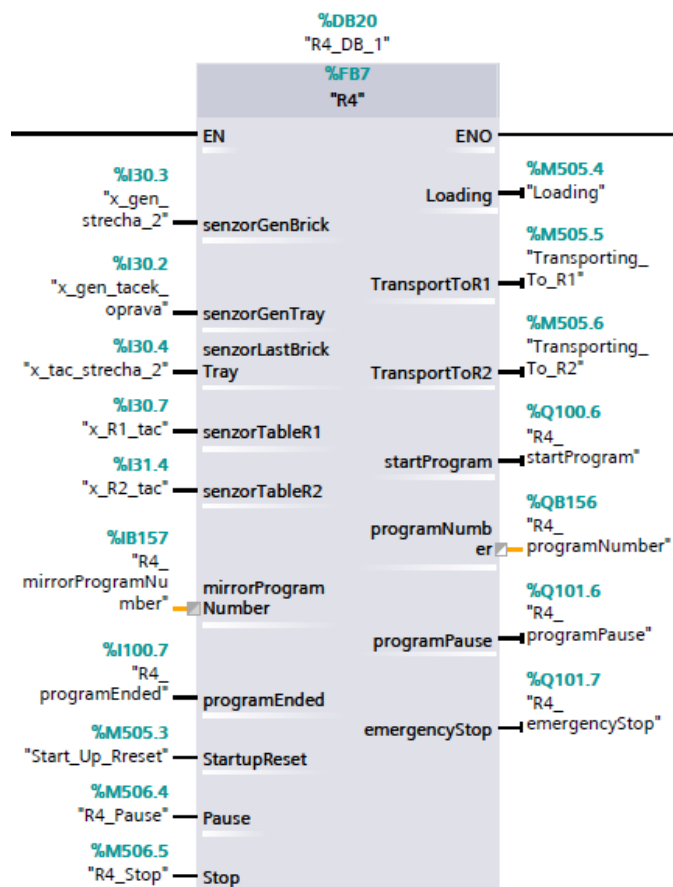
Vyslané číslo rameno pošle zpět přes signál *R1\_mirrorProgramNumber*, když se tato dvě čísla rovnají, signálem *R1\_startProgram* se spustí robotická operace. Vyslané číslo operace je přepočítané z hodnoty čítače, který je uvnitř funkčního bloku a hodnoty počtu operací zapsané na vstupu bloku (vstup *n1stOp*). Jakmile skončí robotická operace, signál *programEnded* zvýší hodnotu čítače o jednu. Proces se opakuje vysláním a přijetím robotického čísla operace, spuštěním a ukončením operace a inkrementací čítače. Proces se opakuje do té doby, než je splněna maximální hodnota čítače, která je zapsána na vstupu bloku *nOfOperations*. Hodnota odpovídá počtu operací, kterou musí robotické rameno provést, aby byl výrobek kompletní. Čítač se vynuluje, když je umístěna poslední část boudy, kterou zachytí senzor. Počet operací pro R1 a R2 je roven pěti a pro R3 je operace jedna.

Vstup pro snímání tácu s materiálem *xMaterial* má nastavenou výchozí hodnotu *true*. Tato hodnota je nastavena z důvodu ramene R3, ke kterému se nepřepřavuje materiál na paletce, proto se zde senzor pro snímání tácu nenachází. Jakmile se na tento vstup nepřipojí žádná proměnná, je vstup v hodnotě *true*. Po připojení proměnné se vstup řídí podle proměnné.

Robotické rameno R4 má svůj vlastní funkční blok (*R4*) zobrazený na obrázku 30. Proces posílání čísel operací funguje úplně totožně jak u funkčního bloku *Rx* obrázek 29. Číslo první robotické operace je spolu s číslem počtu operací uloženy ve funkčním bloku, nijak se nevolí na vstupu funkčního bloku, tak jak je u *Rx*. Podmínkou spuštění robotických operací pro uložení materiálu na ták (40 - 44) jsou vygenerovaný materiál, snímán je pouze druhý střešní díl a ták na pozici pro náklad, zároveň nesmějí být části boudy již naloženy na ták. Všechny čísla operací a jejich popis jsou vypsány v tabulce 4. Jakmile jsou naloženy všechny části, poslední umístěná část (druhý střešní díl) je zachycena senzorem *x\_gen\_tacek\_oprava*, provede se transport naložených částí výrobku (operace 45 nebo 46). Robotické pracoviště R1 má před R2 větší prioritu přepravy nákladu, protože první paletka po spuštění simulace okamžitě zajíždí k prvnímu pracovišti. Během přepravy na pracoviště funkční blok blokuje přes proměnnou spuštění robotického ramene, ke kterému přepravuje materiál.



Obrázek 29: Funkční blok pro robotická ramena R1-R3

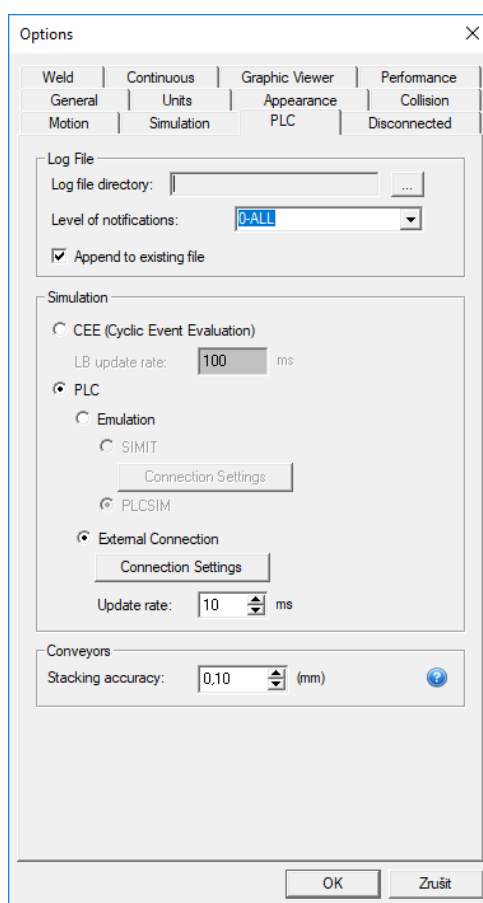


Obrázek 30: Funkční blok pro robotické rameno R4

## 5 Připojení digitálního dvojčete

Projekty vytvořené v TPS lze řídit několika způsoby.

- **Cyclic Event Evaluation (CEE)** - Interní možnost řízení. Skládá se z logických bloků, modulů a větvením jednotlivých operací. Toto řízení nebylo v práci využito pro celou linku.
- **Emulation** - Připojení softwarových simulací
- **External connection** - Možnost externího připojení PLC nebo simulovaného PLC za použití PLCSIMAdvanced. Fyzické PLC lze připojit za použití OPC DA nebo OPC UA serverů. Tyto možnosti připojení budou popsány v této kapitole.



Obrázek 31: Možnosti řízení v TPS

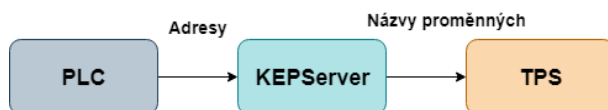
### 5.1 OPC

OPC (Open Platform Communications) vyvinutý OPC Foundation se využívá pro bezpečnou a spolehlivou výměnu dat mezi různými výrobci zařízení v průmyslové automatizaci. Systémy,

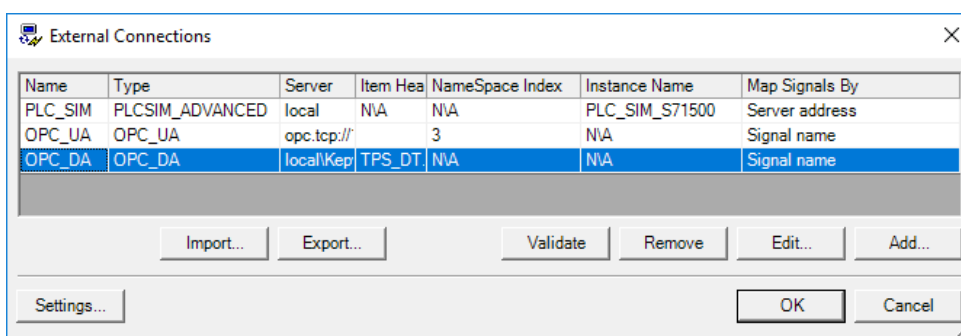
které můžou využívat OPC komunikaci, jsou např. HMI/SCADA, různé databáze a archivace dat. [8]

### 5.1.1 OPC DA

OPC DA neboli OPC Data Access je server ve formě aplikace pro OS Windows. Výměna dat probíhá přes sériový port. OPC server je poté schopen nabízet data jiným aplikacím. Tyto aplikace jsou z pohledu serveru brány jako klienti. Pro vytvoření OPC DA severu byl zvolen program KEPServerEX verze 6. KEPServerEX je ke stažení zdarma, ale tato stažená verze pracuje pouze 2 hodiny, poté je nutné PC restartovat. Přenos dat mezi PLC a KEPServerem je pomocí adres, kde adresy musí sedět a odpovídat těm nahraným v PLC. Přenos dat mezi KEPServerem a TPS je pomocí názvů proměnných. [9]



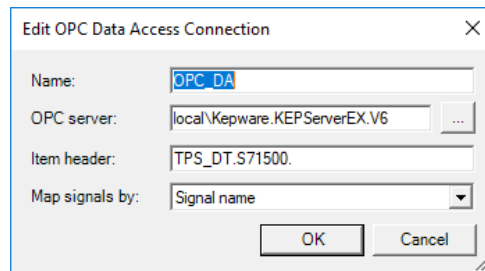
Obrázek 32: Průběh komunikace mezi TPS a PLC



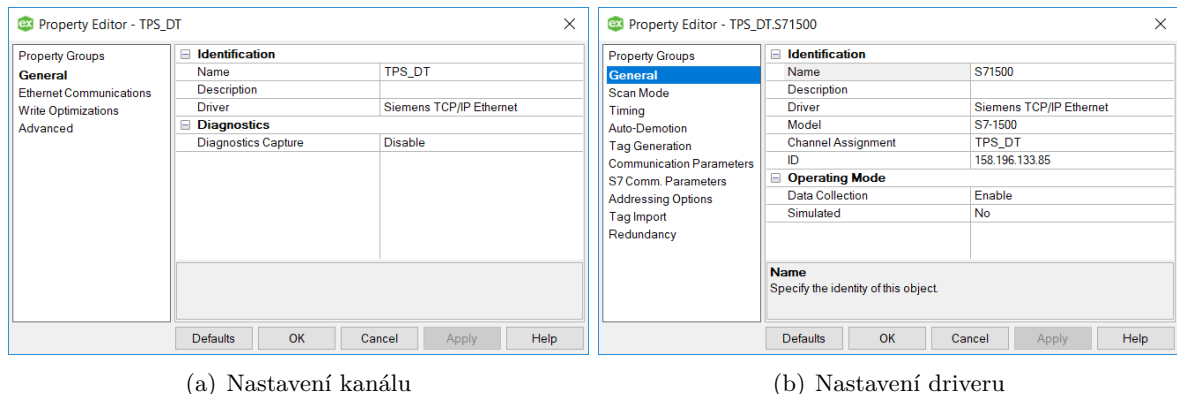
Obrázek 33: Externí připojení

Prvním krokem pro propojení TPS a PLC za použití OPC DA komunikace je vytvoření serveru. V aplikaci KEPServerEx se vytvoří nový kanál, jehož název bude použit při propojení s TPS. Nastaví se odpovídající driver a ethernetové připojení. Dále se vytvoří zařízení, do kterého se zvolí typ a adresa PLC. Název vytvořeného zařízení je také použit v TPS. Název pro tuto práci byl zvolen pro kanál *TPS\_DT* a pro zařízení *S71500*. Driver byl zvolen Siemens TCP/IP Ethernet. Nastavení zařízení bylo uskutečněno podle připojeného PLC S71500 s IP adresou 158.196.133.85. Na obrázku 35 je zobrazeno nastavení kanálu a zařízení.

V TPS po vybrání externího ovládání se musí připojení nakonfigurovat. Na obrázku 34 lze vidět okno pro nastavení komunikace. V okně pro název lze zadat libovolný název pro danou komunikaci. V kolonce zadat adresu OPC severu, která byla *local\Kepware.KEPServerEX.V6*. V okně *Item header* je potřeba zapsat název kanálu a zařízení. Je důležité za každý název napsat tečku. Pro finální propojení je potřeba nastavit signály pro externí připojení.



Obrázek 34: Nastavení OPC DA



(a) Nastavení kanálu

(b) Nastavení driveru

Obrázek 35: Nastavení KEPServeru

### 5.1.2 OPC UA

OPC Unified Architecture (OPC UA) je nový průmyslový komunikační standard vyvinutý OPC Foundation. Na rozdíl od původní specifikace OPC, která je založena na technologii COM/D-COM firmy Microsoft (a tudíž funguje pouze pod OS Windows), jde o technologii založenou na obecně používaných komunikačních standardech, jako jsou TCP/IP, HTTP a SOAP. To znamená, že OPC UA může fungovat i na jiných platformách než Windows. OPC UA komunikaci je možné zabudovat i do vlastních PLC automatů a jiných zařízení.[10]

K vytvoření OPC UA komunikace je potřeba PLC, které jej podporuje. Tuto podmínku splňuje PLC od firmy Siemens Simatic S7-1500 s CPU 1516-3 PN/DP s firmware verzí 2.1. V konfiguraci PLC se musí opc komunikace zapnout. Po zapnutí komunikace se vytvoří adresa komunikace, která byla *opc.tcp://158.196.133.85:4840*, komunikace nesmí být zabezpečena.

Připojení ke komunikaci se nepovedlo, přestože vytvořenou komunikaci TPS našel, ale TPS vypsal chybové hlášení. Na starší verzi TPS (14.0.2) se mi podařilo testovací projekt externě propojit s PLC přes OPC UA komunikaci. Z novější verze (14.1) nešel vytvořený projekt přenést do té starší. Přenos dat mezi PLC a TPS probíhá přes názvy proměnných, ale je potřeba v TPS přepsat proměnné do uvozovek.

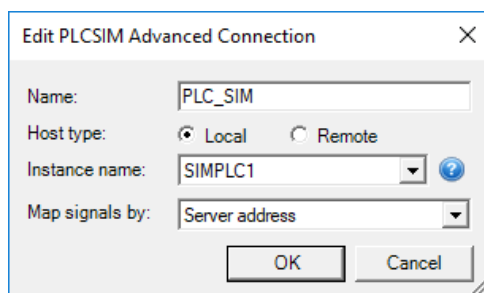


## 5.2 PLCSIM Advanced

PLCSIM Advanced lze použít pro simulování PLC, byla použita verze 2.0 SP1. TPS nabízí možnost se připojit k tomuto simulovanému PLC. Možnost vytvořit až 16 instancí PLC v PLCSIM Advanced. Instance lze vytvářet na různých zařízeních v síti. Lze simulovat PLC kód, HMI a další software pro simulaci výrobních procesů. Další z možností PLCSIM Advanced je simulování OPC UA komunikace. [11]

Nastavení PLCSIM Advanced lze vidět na obrázku 37. V možnosti TCP/IP komunikace je vybrána možnost *local* z důvodu simulování na jenom PC. Název instance lze zvolit jakýkoli. IP adresa byla zvolena podle konfigurace, která byla vytvořena v TIA portálu pro fyzické PLC (158.196.133.85). Kód se kompiluje do simulovaného PLC přes PLCSIM virtual ethernet. V TIA Portálu je potřeba v projektu povolit simulování funkčních bloků.

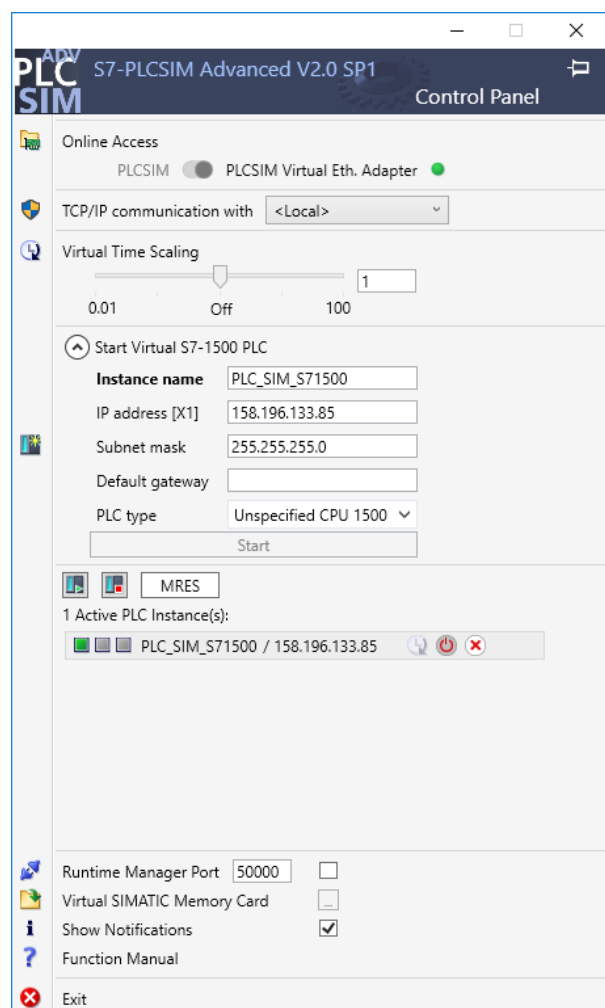
Nastavení v TPS je zobrazeno na obrázku 36. Název při vytváření připojení v TPS je volitelný. Host type se zvolí lokální, jelikož se simulované PLC na stejném počítači. Název instance se zobrazí sám, jestliže je simulace zapnutá. Propojení signálů z TPS a TIA Portálu probíhá přes jejich adresy. Ty se musí přidat v seznamu signálů v TPS, ve sloupci adresy se zapíše číslo adresy a TPS už sám přiřadí, jestli se jedná o výstup nebo vstup. Adresy musí odpovídat těm v TIA portálu.



Obrázek 36: Připojení k PLCSIM Advanced

## 5.3 Nastavení připojení

Jakmile je vytvořena komunikace pomocí které chceme připojit simulaci, musí se nastavit signály, které jsou potřeba pro řízení. V seznamu všech signálů v TPS lze zaškrtnout možnost *PLC Connection*. Zaškrtnutím *PLC Connection* TPS automaticky vypne veškeré řízení v modulech a signál je řízen pouze z externích připojení. Ve sloupečku *External Connection* lze vybrat vytvořené externí připojení. Na obrázku 38 jsou zobrazeny signály, které jsou řízeny externě.



Obrázek 37: Nastavení PLCSIM Advanced

Signal Name	Memory	Type	Address	IEC Format	PLC Connection	External Connection	Resource
Conv_gen_Start	<input type="checkbox"/>	BOOL	10.0	Q10.0	<input checked="" type="checkbox"/>	PLC_SIM	Conv_gen
Conv_gen_Stop	<input type="checkbox"/>	BOOL	10.1	Q10.1	<input checked="" type="checkbox"/>	PLC_SIM	Conv_gen
Conv_1_Start	<input type="checkbox"/>	BOOL	10.2	Q10.2	<input checked="" type="checkbox"/>	PLC_SIM	Conv_1
Conv_1_Stop	<input type="checkbox"/>	BOOL	10.3	Q10.3	<input checked="" type="checkbox"/>	PLC_SIM	Conv_1
Conv_2_Start	<input type="checkbox"/>	BOOL	10.4	Q10.4	<input checked="" type="checkbox"/>	PLC_SIM	Conv_2
Conv_2_Stop	<input type="checkbox"/>	BOOL	10.5	Q10.5	<input checked="" type="checkbox"/>	PLC_SIM	Conv_2
Conv_3_Start	<input type="checkbox"/>	BOOL	10.6	Q10.6	<input checked="" type="checkbox"/>	PLC_SIM	Conv_3
Conv_3_Stop	<input type="checkbox"/>	BOOL	10.7	Q10.7	<input checked="" type="checkbox"/>	PLC_SIM	Conv_3
Conv_4_Start	<input type="checkbox"/>	BOOL	11.0	Q11.0	<input checked="" type="checkbox"/>	PLC_SIM	Conv_4
Conv_4_Stop	<input type="checkbox"/>	BOOL	11.1	Q11.1	<input checked="" type="checkbox"/>	PLC_SIM	Conv_4
Conv_4_test_Start	<input type="checkbox"/>	BOOL	11.2	Q11.2	<input checked="" type="checkbox"/>	PLC_SIM	Conv_4_test
Conv_4_test_Stop	<input type="checkbox"/>	BOOL	11.3	Q11.3	<input checked="" type="checkbox"/>	PLC_SIM	Conv_4_test
Conv_5_Start	<input type="checkbox"/>	BOOL	11.4	Q11.4	<input checked="" type="checkbox"/>	PLC_SIM	Conv_5
Conv_5_Stop	<input type="checkbox"/>	BOOL	11.5	Q11.5	<input checked="" type="checkbox"/>	PLC_SIM	Conv_5

Obrázek 38: Nastavení signálů

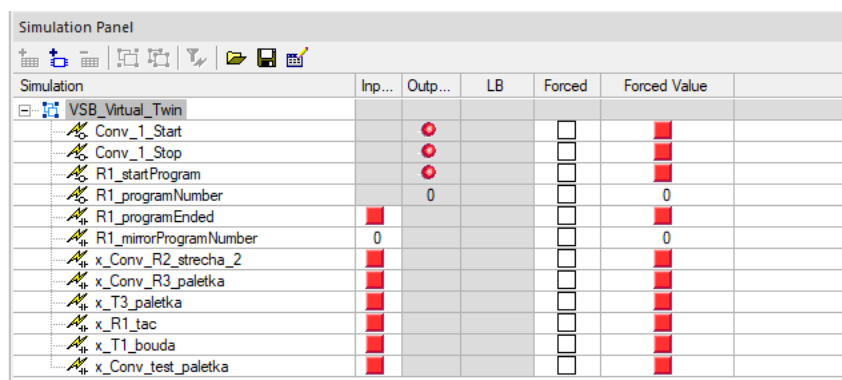
## 6 Testování

Během testování vytvořené řídicí aplikace je potřeba zkontrolovat, zda jsou veškeré proměnné propojeny s digitálním dvojčetem. Jednotlivé druhy komunikací nabízí různé možnosti otestování propojených proměnných. Poté lze bez větších problémů testovat vytvořenou řídicí aplikaci. Během testování je mnohem jednodušší odhalit nedostatky díky možnosti pozorovat chod linky v digitální podobě.

U OPC DA a UA komunikace TPS vypíše hlášku, jestliže nemá přístup k proměnné, která je zvolena pro externí řízení. Bohužel při připojení k simulovanému PLC za použití PLCSIM Advanced TPS nevypíše žádnou hlášku, když adresy nesedí.

V KEP ServeruEx lze zapnout klienta, ve kterém lze vidět veškeré proměnné a jejich stav. U každé proměnné je popis kvality připojení k PLC, jestliže daná proměnná je propojena s proměnnou v PLC.

TPS taky nabízí možnost mapovat stavy proměnných. Na obrázku 39 je simulation panel, do kterého jsou přidány signály, které je potřeba sledovat. V tabulce lze vidět jejich stavy, nabízí také možnost změnit jejich stavy.



Simulation	Inp...	Outp...	LB	Forced	Forced Value
VSB_Virtual_Twin					
Conv_1_Start		●		<input type="checkbox"/>	1
Conv_1_Stop		●		<input type="checkbox"/>	1
R1_startProgram		●		<input type="checkbox"/>	1
R1_programNumber		0		<input type="checkbox"/>	0
R1_programEnded	1			<input type="checkbox"/>	1
R1_mirrorProgramNumber	0			<input type="checkbox"/>	0
x_Conv_R2_strecha_2	1			<input type="checkbox"/>	1
x_Conv_R3_paletka	1			<input type="checkbox"/>	1
x_T3_paletka	1			<input type="checkbox"/>	1
x_R1_tac	1			<input type="checkbox"/>	1
x_T1_bouda	1			<input type="checkbox"/>	1
x_Conv_test_paletka	1			<input type="checkbox"/>	1

Obrázek 39: Simulation panel

Během testování bylo odladěno mnoho chyb. Jedním z chyb byly špatně vytvořené proměnné v serveru, špatně navržené algoritmy pro řízení a mnoho dalších. Prvotní testování aplikace bylo prováděno s použitím OPC DA komunikace, kde je potřeba vytvořit server, tvorba serveru není nijak náročná, ale je potřeba dávat si pozor na názvy a adresy, a také zabere dost času. Použitím simulovaného PLC je mnohem jednodušší začít s testováním dříve, protože není potřeba vytvářet server, stačí aby seděly adresy u signálů, které je potřeba využít pro řízení. Při spuštění simulaci PLC, nebylo zaznamenáno nějak výrazné snížení výkonu jak simulovaného PLC a PC. Použitím OPC UA komunikace se musí názvy proměnných, které chceme propojit shodovat a v TPS se musí přidat to uvozovek.

Výsledkem testování jsou videa, která se nachází v příloze. Jsou nahrány robotické operace všech ramen a celkový záběr na linku.

## 7 Závěr

Cílem bakalářské práce bylo navrhnout a realizovat řídicí aplikaci pro PLC za použití digitálního dvojčete. Aplikace řídí vytvořené digitální dvojče v programu Tecnomatix Process Simulate (TPS). Připojení digitálního dvojčete k fyzickému PLC bylo realizováno za použití OPC komunikací a simulovaného PLC.

V první části této práce byla provedena analýza možností návrhů a testování řídicí aplikace za použití digitálního dvojčete. Z analýzy byl zvolen V-model, který je rozdělen na dvě strany. Levá strana se zabývá analýzou a návrhem řídicího systému a pravá strana samotným testováním návrhů z levé strany. Dále je popsána problematika digitálního dvojčete, vypsán teoretický postup tvorby digitálního dvojčete a vypsány výhody použití dvojčete během testování řídicí aplikace.

V druhé kapitole je popsáno vytvořené digitální dvojče, které vzniklo ve spolupráci na projektu Smart Factory ve škole, které bylo vytvořené v softwaru TPS od firmy Siemens. Digitální dvojče se skládá ze čtyř robotických ramen, z toho jedno je umístěné na pojezdu a soustavy dopravníků. Dvě robotická ramena skládají výrobek, třetí provádí testování a čtvrté nakládá a dopravuje materiál k pracovištím. Robotická ramena skládají výrobek z LEGO kostek, které jsou propojeny do bloků stěn a střešních částí. Spojeny jsou z důvodu ušetření robotických operací. Výrobek je skládán na paletku, která je dopravovaná pomocí soustavy dopravníků k pracovištím a odběrnému místu. Vytvořený materiálový tok zajistí, že vygenerovaný materiál projde operacemi naložení, sestavení, otestování a ukončení materiálového toku probíhá na odběrném dopravníku, tímto je vytvořen životní cyklus produktu. Dále je vypsán postup vytvoření pojezdu, který byl vytvořen pro čtvrté rameno. V kapitole jsou vypsány robotické operace, které jsou potřeba pro vytvoření funkční řídicí aplikace a popis jak funguje řízení dopravníků a robotických ramen.

Další část práce se zabývá samotným návrhem řídicí aplikace pro PLC. Pro návrh je vytvořena funkční analýza, která popisuje jak mají fungovat jednotlivé části linky. Funkční analýza popisuje čtyři dopravníkové křižovatky, jak se mají na nich paletky pohybovat pomocí dopravníků, které se na křižovatkách nacházejí. Dále popisuje princip řízení ramen za jakých podmínek se můžou spustit, postup volání operací a kdy má být přepravován materiál. Výstupem z této kapitoly jsou vstupní a výstupní signály, které jsou vypsány v tabulkách s popisem k čemu signál slouží.

Čtvrtá kapitola popisuje vytvořenou řídicí aplikaci, která je napsána v jazyce LAD (ladder). Realizována je v programu TIA Portál v14. Pro řízení bylo zvoleno PLC S7-1500 od firmy Siemens. V kapitole jsou popsány jednotlivé funkční bloky a jak v nich fungují vytvořené algoritmy. Funkční bloky jsou vytvořené pro jednosměrné a obousměrné dopravníky, robotická ramena, testovací stanoviště a řízení provozu na dopravníkových křižovatkách. Bloky jsou vytvořeny tak, aby se dal funkční blok využít vícekrát pro danou část linky. Například jednosměrných dopravníků je v systému třináct, každý z těchto dopravníků má stejný funkční blok, ale s jiným datovým

blokem. Ramena, která nemají pojezd, sdílejí také stejný funkční blok, ale každé rameno má svůj datový blok. Čtvrté robotické rameno má vlastní funkční blok, kvůli pojezdu, který je potřeba také řídit.

V páté kapitole jsou popsány a otestovány jednotlivé možnosti komunikací mezi PLC a digitálním dvojčetem. Otestovány byly komunikace OPC DA a OPC UA. Pro vytvoření serveru potřebného k propojení PLC a dvojčete byl zvolen software KEPServerEX V6. Zvolené PLC, pro které byla vytvořena hardwarová konfigurace podporuje OPC UA komunikaci. Poslední část kapitoly popisuje propojení TPS se simulovaným PLC. Instance simulovaného PLC byla vytvořena v programu PLCSIM ADVANCED v2. Každý ze způsobu připojení dvojčete k PLC je důkladně popsán. Z otestovaných možných připojení mi nejvíce vyhovuje použití simulovaného PLC.

Poslední kapitola se věnuje testování řídicí aplikace. V kapitole jsou popsány možnosti ověření správného propojení proměnných s dvojčetem a PLC. Během této fáze bylo odstraněno mnoho chyb. K opravení chyb napomohlo dvojče, díky kterému je množnost pozorovat chod linky v digitálním zprovoznění. Výsledkem testování jsou videa, která jsou v příloze.

## Literatura

- [1] Digitální dvojče [online]. Česká republika: Siemens, s.r.o [cit. 201812-21]. Dostupné z: <https://www.siemens.cz/industryforum/digitalni-dvojce>
- [2] Grieves, Michael. (2016). Origins of the Digital Twin Concept. 10.13140/RG.2.2.26367.61609.
- [3] Alam, Kazi Masudul & El Saddik, Abdulmotaleb. (2017). C2PS: A Digital Twin Architecture Reference Model for the Cloud-based Cyber-Physical Systems. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2657006.
- [4] Qi, Qinglin & Tao, Fei & Zuo, Ying & Zhao, Dongming. (2018). Digital Twin Service towards Smart Manufacturing. Procedia CIRP. 72. 10.1016/j.procir.2018.03.103.
- [5] SDLC V-Model [online]. [cit. 2019-01-18]. Dostupné z: [https://www.tutorialspoint.com/sdlc/sdlc\\_v\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)
- [6] Dokumentace Siemens: Tecnomatix eMServer Platform 14.1 Help [online]. Siemens, 2018 [cit. 2019-04-22]. Dostupné z: [https://docs.plm.automation.siemens.com/tdoc/tecnomatix/14.1/tecnomatix\\_eMS/#uid:-index](https://docs.plm.automation.siemens.com/tdoc/tecnomatix/14.1/tecnomatix_eMS/#uid:-index)
- [7] General Tecnomatix Forum [online]. Siemens, 2019 [cit. 2019-04-22]. Dostupné z: <https://community.plm.automation.siemens.com/t5/General-Tecnomatix-Forum/bd-p/tecnomatix-forum>
- [8] Komunikace přes rozhraní OPC [online]. Ostrava: MICROSYS, spol. s r.o. [cit. 2019-04-22]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPC.htm>
- [9] Komunikace přes rozhraní OPC DA [online]. Ostrava: MICROSYS, spol. s r.o. [cit. 2019-04-22]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPCDA.htm>
- [10] Komunikace přes rozhraní OPC UA [online]. Ostrava: MICROSYS, spol. s r.o. [cit. 2019-04-22]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPCUA.htm>
- [11] PLCSIM Advanced [online]. Siemens, 2018 [cit. 2019-04-23]. Dostupné z: [https://w5.siemens.com/web/cz/cz/corporate/portal/home/produkty\\_a\\_sluzby/IADT/ti\\_a\\_na\\_dosah/Documents/2018\\_duben/softwarevy-nastroj-SIMATIC-PLCSIM-Advanced.pdf](https://w5.siemens.com/web/cz/cz/corporate/portal/home/produkty_a_sluzby/IADT/ti_a_na_dosah/Documents/2018_duben/softwarevy-nastroj-SIMATIC-PLCSIM-Advanced.pdf)

## A Seznam příloh na přiloženém CD

- Vytvořená linka v Tecnomatix Process Simulate v14.1, která je uložena v archivovaném souboru.rar s názvem VSB\_Virtual\_Twin. Nutné vytvořit cestu pro projekt C:\PLM\Process\_Simulate\_data\Projects a do složky Projects rozbalit soubor, cesta musí odpovídat, jinak se projekt neotevře správně.
- Řídící aplikace, vytvořený v TIA Portálu v14. Složka s názvem PLC\_aplikace
- Video ukázky nahrané při propojení dvojčete s reálným PLC. Video se nachází ve složce Video.
- Bakalářská práce ve formátu VIC0063.pdf